

**Entwurf gebrauchstauglicher
Darstellungsformen für eine effiziente
Vermittlung von Ereignissen in einer
verteilten Simulationsumgebung unter
Berücksichtigung nutzerzentrierter
Ansätze**

Masterarbeit

im Studiengang
Medieninformatik

vorgelegt von

Magnus Szaffenauer

Matr.-Nr.: 11068173

am 15.04.2015

an der Fachhochschule Köln

Erstprüfer/in:	Prof. Dr. Prof. Gerhard Hartmann
Zweitprüfer/in:	M.Sc. Oliver Seebach

Danksagung

Die Anfertigung dieser Arbeit wurde durch eine Unterstützung der Einrichtung „Software und Simulationstechnik des Deutschen Zentrum für Luft und Raumfahrt e.V.“ ermöglicht. Mein besonderer Dank geht daher an meine Betreuer, Herrn Oliver Seebach sowie Frau Doreen Seider und an die freundlichen sowie hilfsbereiten Kollegen der Einrichtung. Bei offenen Fragestellungen, wurde mir in jeder Situation engagiert weitergeholfen. Vielen Dank für diese Unterstützung bei der Durchführung meiner Arbeit und das überaus freundliche und hilfsbereite Umgehen miteinander.

Ebenfalls geht mein Dank auch meinen Betreuer der Fachhochschule Gummersbach, Herrn Gerhard Hartmann, der mir bei meiner Arbeit mit hilfreichen Ratschlägen zur Seite stand und stets bemüht war, mich bei meiner Arbeit zu unterstützen.

Des Weiteren, möchte ich meinen Dank an die Probanden richten, die sich im Rahmen einer Nutzungskontextanalyse sowie der abschließenden Evaluation die Zeit genommen haben, mich bei meiner Arbeit zu unterstützen.

Auch möchte ich allen danken, die mich während der Anfertigung meiner Masterarbeit im Alltag begleitet und unterstützt haben. Daher bedanke ich mich insbesondere bei meiner Familie und meinen Freunden, die mich jederzeit motiviert haben.

Inhaltsverzeichnis

Danksagung	2
Inhaltsverzeichnis	3
Abbildungsverzeichnis.....	6
1 Einleitung	8
1.1 Problemstellung und Forschungsfrage	8
1.2 Vorangegangene Arbeiten	10
1.3 Zielsetzung	11
2 Methodologie	13
2.1 Analysephase	13
2.1.1 Menschliche Kognition und Aufmerksamkeit eines Menschen	13
2.1.2 Awareness.....	14
2.1.3 CSCW-Systeme	14
2.1.4 Darstellungsformen für Benachrichtigungen	14
2.2 Konzeptphase	15
2.3 Entwicklung von Gestaltungslösungen	16
2.4 Evaluation der Gestaltungslösungen	17
2.5 Iterative Überarbeitung.....	18
3 Remote Component Environment	20
3.1 Das System	20
3.1.1 Workflows	21
3.1.2 Tools	21
3.1.3 Instanzen in RCE.....	22
3.2 Darstellung von Ereignissen in RCE	23
3.2.1 Network-View	23
3.2.2 Workflow Data Browser-View.....	23
3.2.3 Log-View.....	24
3.2.4 Workflow List-View	25
3.2.5 Workflow Console-View.....	25
3.2. Ereignisse in RCE.....	26
3.2.1. Tool-Ereignisse	27
3.2.2. Workflow-Ereignisse	27
3.2.6 System-Ereignisse	28
4 Analyse relevanter Konzepte und Begrifflichkeiten	30

4.1. Relevante Konzepte	30
4.1.1 CSCW-Systeme	30
4.1.2 Aufmerksamkeit	33
4.1.3 Awareness.....	37
4.1.4 Awareness in einer verteilten Simulationsumgebung	39
4.2 Ereignisse in einer verteilten Simulationsumgebung	40
4.2.1 Taxonomie nach McGrenere	41
4.2.2 Anpassung der Taxonomie	42
4.2.3 Klassifikation der Ereignisse	43
4.3 Benachrichtigungen in einer verteilten Simulationsumgebung.....	45
4.3.1 Begriffsdefinition – Benachrichtigung	45
4.3.2 Darstellungsformen für Benachrichtigungen	46
4.3.3 Klassifikation der Benachrichtigungen	56
4.3.4 Das Modell der Ereignisse und Benachrichtigungen	57
5 Konzept.....	63
5.1 Tagging-System.....	63
5.1.1 Tags	64
5.1.2 Tag-Interaktion	64
5.2 Notification Center	66
5.2.1 Notification-Groups.....	67
5.2.2 Syntax einer Ereignis-Benachrichtigung	69
5.2.3 Notification-Fokus.....	70
5.2.4 Notification-Cluster	73
5.2.5 Individualisierbarkeit.....	75
5.2.5.3 Individualisierbarkeit der Notification-Center-Benutzeroberfläche.....	79
5.3 Notification-Bar.....	80
5.4 Workflow Trigger-Events	83
5.4.1 Trigger	83
5.4.2 Logische Verknüpfung von Triggern	84
5.5 Events	87
5.5.1 Anlegen eines Trigger-Events	88
5.6 Tool-Verfügbarkeit.....	90
5.6.1 Komponenten-Indikatoren.....	91
5.6.2 Darstellung von Komponenten-Zuständen.....	92
6 Evaluation	95
6.1 Probleme und Risiken.....	95
7.1.1. Ressourcen bei der Evaluation.....	96
7.1.2. Evaluation von Awareness-Aspekten	98
6.1.2 Evaluation einer Anpassung eines bestehenden Konzeptes	98
6.1.3 Erstellung realitätsnaher Tasks.....	99

6.2	Metriken der Evaluation	100
6.3	Empirische Evaluation – Konzept	101
6.3.1	Beobachtungen	102
6.3.2	Befragungen	103
6.4	Analytische Evaluation	106
6.4.1	Cognitive Walkthrough	107
6.4.2	Groupware Walkthrough	109
6.5	Pluralistic Walkthrough	111
7	Planung der Evaluation	114
7.1.1	Definition der Tasks	114
7.1.2	Definieren geeigneter Fragestellungen	116
7.2	Bewertung und iterative Überarbeitung	118
7.2.1	Iterative Anpassung des Notification-Centers	118
7.2.2	Iterative Anpassung der Trigger-Events	119
8	Fazit	122
8.1	Ausblick	123
Anhang A: Task-Descriptions		125
Anhang B: Balsamiq-Mockups		128
A.1	Notification-Center	128
A.2	Trigger-Events	139
Anhang B: Paper-based Mockups		143
Quellenverzeichnis		146
Eidesstattliche Versicherung		148
Auszug aus dem Strafgesetzbuch (StGB)		148

Abbildungsverzeichnis

Abbildung 1: Editor des Axure Mockup-Tools	17
Abbildung 2: Darstellung eines Workflows in RCE	21
Abbildung 3: Network-View von RCE mit Darstellung der RCE-Instanzen	23
Abbildung 4: Workflow Data Browser des RCE-Systems	24
Abbildung 5: Log-View in RCE mit Darstellung der grafischen Indikatoren	25
Abbildung 6: Workflow List-View in RCE	25
Abbildung 7: Workflow Console-View in RCE	26
Abbildung 8: Modell der Aufmerksamkeit nach Broadbent	34
Abbildung 9: Aufmerksamkeits-Nutzen Tabelle nach McCrickard und Chewar	36
Abbildung 10: Mögliche Gliederung der user notification goals	37
Abbildung 11: Ballon-Message unter Windows 7	47
Abbildung 12: Warnmeldung in Form eines Popups in RCE	48
Abbildung 13: Darstellung eines Tool-Tips unter Windows	49
Abbildung 14: Darstellung eines Notification-Centers unter iOS 6	50
Abbildung 15: Notification-Bar unter iOS	51
Abbildung 16: Darstellung von Tool-Zuständen in RCE. t	52
Abbildung 17: Mail-Kopf mit Statusdarstellung in MS Outlook	52
Abbildung 18: Notification-Interfaces nach Booker et.al	54
Abbildung 19: Scope Notification System nach Dantzig et.al.	55
Abbildung 20: Modell mit Klassifizierung der Ereignis-Initiatoren	60
Abbildung 21: Gestaltungslösung der Tag-Darstellung	64
Abbildung 22: Gestaltungslösung eines Dialoges zur Erstellung eines neuen Tags	65
Abbildung 23: Kontextmenü mit Einträgen des Tagging-Systems	66
Abbildung 24: Gestaltungslösung der Benachrichtigungs-Syntax	70
Abbildung 25: Gestaltungslösung des Notification-Centers mit Notification-Fokus	73
Abbildung 26: Gestaltungslösung geschlossener Notification-Cluster	74
Abbildung 27: Gestaltungslösung eines geöffneten Notification-Clusters	75
Abbildung 28: Einstellungen des Notification-Centers anhand eines Prototypens	77
Abbildung 29: Gestaltungslösung für das Anlegen einer neuen Notification-Group	79
Abbildung 30: Gestaltungslösung einer Benachrichtigung in der Notification-Bar	81
Abbildung 31: Gestaltungslösung der Notification-Counter Darstellung	82
Abbildung 32: Gestaltungslösung der Trigger-Event-Einstellungen. Über die "add Trigger" Funktionalität können weitere Trigger durch eine logische Verknüpfung hinzugefügt werden.	86
Abbildung 33: Papier-basierte Gestaltungslösung für die Darstellung von Tool- Zuständen Die Indikatoren bestehen dabei aus dem eigentlichen Icon der jeweiligen Komponente, sowie aus einem Symbol, welches Informationen über die Verfügbarkeit vermitteln soll.	92
Abbildung 34: Darstellung der Zustände von Tools im Notification-Center	93

1 Einleitung

Die Entwicklung von Software-Systemen, die sich mit der Bewältigung von komplexen Aufgaben bzw. Problemstellungen unterschiedlichster Forschungsgebiete befassen, stellt eine Vielzahl an Herausforderungen an die Forschung der Mensch-Computer-Interaktion dar. So erfordern komplexe Aufgaben oftmals den Einsatz interdisziplinären Wissens und spezifische Fähigkeiten von Forschern sowie Ingenieuren, welches sich nur durch ein kollaboratives Aufgabenumfeld effektiv aggregieren lässt. Mitunter ist es jedoch aufgrund diverser Randbedingungen, wie physischer Distanz oder mangelnder Zeit nicht möglich, solche kollaborativen Umfelder zu realisieren. Mit der Etablierung des Internets, zuverlässigen Netzwerkarchitekturen und neuer Sicherheitsmechanismen, erlangt daher auch die Entwicklung verteilter Software-Systeme zunehmend Aufmerksamkeit. Durch den Einsatz verteilter Software-Systeme, lassen sich komplexe Aufgaben und Problemstellungen kollaborativ bewältigen, ohne an physische Randbedingungen gebunden zu sein. Das Forschungsgebiet der Mensch-Computer-Interaktion hat hierbei insbesondere das Ziel, die Entwicklung solcher Systeme mit einem hohen Maß an Gebrauchstauglichkeit zu ermöglichen und eine ganzheitliche Sicht an die Anforderungen solcher Systeme, die je nach Domäne bzw. Einsatzgebiet variieren können, zu bilden. Neben der allgemeinen Konzeption von Aufgabenangemessenen Interaktionsmöglichkeiten für die Nutzer verteilter Systeme, befassen sich verschiedene Teilgebiete der MCI mit einer effizienten Vermittlung von Ereignissen, die bei einer kollaborativen Aufgabenbewältigung eintreten können. (Beispiele? Awareness etc?) Anhand der verteilten Simulationsumgebung RCE (Fußnote), wird im Rahmen dieser Ausarbeitung der Prozess einer Konzeption von gebrauchstauglichen Darstellungsformen von Ereignissen eines verteilten Systems verdeutlicht, und auf spezifische Probleme während der Entwicklung eingegangen.

1.1 Problemstellung und Forschungsfrage

Verteilte Systemen, die eine kollaborative Aufgabenbewältigung durch mehrere Nutzer vorsehen, generieren eine Vielzahl an Ereignissen, die sich wesentlich auf die erfolgreiche Bewältigung einer Aufgabe auswirken können. Während diese Ereignisse in einer physisch eng gekoppelten Umgebung, in der sich alle Gruppenteilnehmer gemeinsam in einem Raum befinden, über kognitive Vorgänge des Menschen unmittelbar wahrgenommen werden können, stellt eine Vermittlung dieser, über ein kollaborativ verteiltes System eine große Herausforderung an die Konzeption gebrauchstauglicher Ereignis-Benachrichtigungen dar. So bedingt der Einsatz eines solchen Systems Restriktionen, die einer effizienten Vermittlung von Ereignissen

entgegenwirken. Zu diesen Restriktionen zählt beispielsweise die nur begrenzt zur Verfügung stehende Fläche eines Displays, auf der alle Ereignis-Informationen und weitere Gruppenkontext-relevante Informationen ganzheitlich und effizient abgebildet werden müssen¹. Bei einer Konzeption gebrauchstauglicher Darstellungsformen wird daher angestrebt, dass auch die Ereignisse in verteilten Software-Systemen so vermittelt werden, dass die Nutzer diese ohne erheblichen Aufwand oder Beeinträchtigung ihrer primären Aufgaben wahrnehmen bzw. verarbeiten können. In der verteilten Simulationsumgebung RCE² kann interdisziplinäres Wissen von Forschern oder Ingenieuren gemeinsam genutzt werden, um Simulationen durchzuführen, anhand derer sich komplexe Systeme (z.B. aus der Luft- und Schifffahrt) analysieren und entwickeln lassen. Dabei treten sowohl lokal- als auch verteilt ausgelöste Ereignisse ein, die den Nutzern in einem hohen Maß an Gebrauchstauglichkeit zu vermitteln sind. Das Problem bei einer Konzeption angemessener Darstellungsformen dieser Ereignisse liegt nun darin, festzustellen welche Anforderungen an die Ereignis-Benachrichtigungen überhaupt zu berücksichtigen sind, damit diese durch den Nutzer ohne erheblichen kognitiven Aufwand verarbeitet werden können. Dieses Problem wird dadurch verstärkt, dass die auftretenden Ereignisse in verteilten Systemen wie RCE, aus verschiedenen Quellen stammen können, die ggf. unterschiedliche Anforderungen an eine angemessene Darstellung erfordern.

Diese Problemstellung führt grundlegend zu der Frage, ob und in wie fern unterschiedliche Ereignisse zu abweichenden Anforderungen an die Konzeption gebrauchstauglicher Darstellungsformen führen. Ist es möglich, grundlegende Anforderungen an Ereignis-Benachrichtigungen eines verteilten Systems zu identifizieren, die sowohl für das in diesem Projekt betrachtete RCE-System, als auch allgemein für verteilte Systeme verschiedener Domänen gleichermaßen gültig sind? Für die Konzeptionierung eines gebrauchstauglichen Benachrichtigungs-Systems, ist die Identifikation der Anforderungen an die Ereignis-Darstellung essenziell, da sich nur auf Basis einer ganzheitlichen Betrachtung der Anforderungen angemessene Gestaltungslösungen erstellen lassen. In dieser Ausarbeitung, soll die soeben erwähnte Frage daher genauer betrachtet und gemäß eines geeigneten Modells zur Festlegung der Anforderungen an Ereignis-Benachrichtigungen beantwortet werden. Auf Basis dieser Anforderung erfolgt anschließend die Konzeption eines Benachrichtigungs-Systems für die verteilte Simulationsumgebung RCE.

¹ Grudin J. 2001, S. 2

² <https://software.dlr.de/rcenvironment/home/> (Letzter Zugriff: 14.04.2015)

Mit der Konzeption gebrauchstauglicher Darstellungsformen für Ereignis-Benachrichtigungen ergibt sich zudem die Problematik, diese anschließend angemessen zu evaluieren. Da die Konzeption der Vermittlung von Ereignissen in einem verteilten System unterschiedliche Domänen der Mensch-Computer-Interaktion, wie der Psychologie, den Sozialwissenschaften oder der „Awareness“³ beinhaltet, ist auch die anschließende Evaluation des Konzeptes vorzubereiten und sollte die relevanten Aspekte dieser Domänen berücksichtigen. Hierbei stellt sich insbesondere die Frage, durch welche Faktoren sich gebrauchstaugliche Ereignis-Benachrichtigungen auszeichnen und wie sich diese angemessen evaluieren lassen. Auf Grundlage einer Identifikation relevanter Faktoren für Ereignisse, sowie der Durchführung einer Evaluation, soll diese Frage in dieser Ausarbeitung ebenfalls beantwortet werden.

1.2 Vorangegangene Arbeiten

Eine Auseinandersetzung mit der in diesem Kapitel angesprochenen Problematik und den daraus resultierenden Forschungsfragen, führte zu einer ausführlichen Recherche nach Arbeiten, die sich bereits mit ähnlichen Problem- sowie Fragestellungen befassen haben. An dieser Stelle sind dabei insbesondere zwei Ausarbeitungen zu erwähnen, deren Fokus ebenfalls auf dem Entwurf gebrauchstauglicher Darstellungsformen für Benachrichtigungen in verteilten Systemen wie beispielsweise CSCW-Systemen⁴ lag. So definiert das „Scope Notification System“ nach Dantzieh et. al.⁵ ein grafisches Benachrichtigungs-System, das Ereignisse aus verschiedenen Kategorien über ein simpel gehaltenes Interface darstellt. Das Konzept richtet sich dabei an keine bestimmte Domäne und sieht, je nach Anforderungen des jeweiligen Kontextes, eine Anpassung der darin abgebildeten Ereignis-Kategorien vor. Anhand des Systems kann der Nutzer beispielsweise Informationen über eingehende E-Mails, Kalendereinträge oder Aufgaben erhalten und soll so ein besseres Bewusstsein über relevante Benachrichtigungen entwickeln können. Der Fokus liegt somit insbesondere auf dem Forschungsgebiet der Awareness über Ereignisse in einem verteilten System.

Ein weiterer Ansatz, der die Konzeption eines Benachrichtigungs-Systems für die kollaborative Entwicklungsumgebung „JAZZ“⁶ vornimmt, wird in dem Paper von Joanna McGrenere et. al. beschrieben⁷. Auf Basis verschiedener deskriptiver Szenarien, identifizieren und priorisieren McGrenere et.al die kritischen Faktoren, die in JAZZ unter anderem zu einem Mangel an Bewusstsein über den Gruppenkontext führen. Zu diesen Faktoren zählt unter anderem das Bewusstsein über Änderungen innerhalb der

³ Die Awareness (dt. Bewusstsein), kann als Teil-Disziplin der Psychologie bezeichnet werden

⁴ Computer-supported-collaborative-work

⁵ Dantzieh M. 2002, S. 268

⁶ <https://jazz.net> (Letzter Zugriff: 22.03.2015)

⁷ McGrenere J. 2009

Gruppe, sowie die Priorisierung eingehender Benachrichtigungen je nach Kontext eines Nutzers. Neben diesen vergleichsweise sehr spezifischen Ausarbeitungen, existieren jedoch auch wissenschaftliche Auseinandersetzungen, die grundlegenden Fragestellungen nach der Gebrauchstauglichkeit solcher Benachrichtigungs-Systeme nachgehen. So beschreibt das Modell aus Aufmerksamkeits-Kosten und Benachrichtigungs-Nutzen nach McCrickard et. al.⁸ welche Faktoren dazu beitragen können, dass ein ineffizientes Verhältnis aus Aufmerksamkeit eines Nutzers, sowie dem Nutzen einer Benachrichtigung entsteht. Eine allgemeine Definition des Awareness-Begriffs, sowie dessen Relevanz in Bezug von verteilten Systemen wird in der Arbeit „The problem with awareness“ von Kjeld Schmidt⁹ vorgenommen. Schmidt verdeutlicht in dieser Arbeit unter anderem die Relevanz des Begriffs für CSCW-Systeme und wirft grundlegende Fragestellungen auf, die eine Sensibilisierung auf dieser Domäne ermöglichen soll. Die in diesem Abschnitt kurz vorgestellten Arbeiten, bezeichnen dabei jedoch nur einen Bruchteil an Auseinandersetzungen auf den als relevant erachteten Forschungsgebieten. Auch wenn die in diesem Projekt vorgestellte Problemstellung sowie die Forschungsfragen sehr spezifisch ausfallen, existiert eine Vielzahl an wissenschaftlichen Ausarbeitungen, die eine ähnliche Thematik behandeln. Eine ausführliche Betrachtung der als relevant erachteten Konzepte aus vorangegangenen Arbeiten, kann der Analysephase dieses Dokumentes entnommen werden.

1.3 Zielsetzung

Gemäß der Problemstellung und den daraus resultierenden Forschungsfragen, besteht die Zielsetzung dieser Ausarbeitung unter anderem darin, eine angemessene Analyse der Anforderungen an Ereignis-Benachrichtigungen in einem verteilten System durchzuführen, um auf deren Grundlage gebrauchstaugliche Darstellungsformen für diese Benachrichtigungen entwickeln zu können. Die Identifikation relevanter Anforderungen soll dabei jedoch nicht ausschließlich für die verteilte Simulationsumgebung RCE gültig sein, sondern sich allgemein für verteilte Systeme aus verschiedenen Domänen gleichermaßen eignen. Für die Konzeption angemessener Gestaltungslösungen, besteht zudem ein Ziel darin, eine Betrachtung relevanter Konzepte aus den Forschungsgebieten der Mensch-Computer-Interaktion vorzunehmen, und diese in der Entwurfsphase nach einer angemessenen Abwägung zu berücksichtigen. Mit der Konzeption eines gebrauchstauglichen Benachrichtigungs-Systems, ergeben sich zudem besondere Ansprüche an eine anschließende Evaluation

⁸ McCrickard D. 2003, S.67

⁹ Schmidt K. 2002

entworfener Gestaltungslösungen, weshalb diese Ausarbeitung auf eine effiziente und möglichst ganzheitliche Vorbereitung sowie Durchführung eines angemessenen Evaluations-Prozesses abzielt. Wie in Kapitel 1.1. angedeutet, müssen für eine angemessene Evaluation zunächst die Faktoren identifiziert werden, anhand derer sich gebrauchstaugliche Benachrichtigungen definieren lassen. Im Rahmen dieser Ausarbeitung soll eine Identifikation dieser Faktoren stattfinden und im Prozess der Evaluation als Metriken eingesetzt werden.

2 Methodologie

Die Konzeption geeigneter Darstellungsformen für Ereignisse in einer verteilten Simulationsumgebung bezeichnet ein sehr spezifisches Projekt, das besondere Anforderungen an die Durchführung eines nutzerzentrierten Ansatzes zur Entwicklung von gebrauchstauglichen Gestaltungslösungen stellt. Dementsprechend muss der Prozess der Analyse, der Konzeption sowie der anschließenden Evaluation angemessen vorbereitet werden, um ein qualitativ hochwertiges Konzept erstellen zu können. In diesem Kapitel, wird die grundlegende Methodologie erläutert, um dem Leser ein Verständnis über den Ablauf, ausgehend von der Forschungsfrage bis hin zu einem angemessenen Konzept, zu vermitteln.

2.1 Analysephase

Die Analyse relevanter Konzepte bezeichnet einen wesentlichen Bestandteil dieses Projektes. Aufgrund des spezifischen Kontextes der Problemstellung, die innerhalb dieser Arbeit behandelt wird, war eine ausführliche Auseinandersetzung auf verschiedenen Domänen der Mensch-Computer-Interaktion notwendig und wurde in dieser Phase vorgenommen. Bei einer Recherche nach wissenschaftlicher Literatur, die sich im Rahmen dieser Arbeit anbieten, wurde so der Fokus unter anderem auf die Aspekte der Aufmerksamkeit, der „Awareness“ oder der kognitiven Verarbeitung von Informationen durch den Menschen gelegt. Insgesamt wurden folgende Forschungsgebiete und Konzepte im Kontext dieser Ausarbeitung identifiziert, deren Relevanz für das Projekt kurz erläutert wird:

2.1.1 Menschliche Kognition und Aufmerksamkeit eines Menschen

Die auftretenden Ereignisse in einem verteilten System, sollen den Nutzern auf eine effiziente und gebrauchstaugliche Art zugänglich gemacht werden. Um dies zu ermöglichen, ist es zunächst erforderlich, ein allgemeines Verständnis über die menschliche Kognition sowie der Aufmerksamkeit eines Menschen zu entwickeln. Die Aufmerksamkeit des Menschen kann allgemein als eine stark eingeschränkte Ressource betrachtet werden, die in der kognitiven Verarbeitung als Filter eingehender Reize eingesetzt werden kann. Ein hinreichendes Verständnis über die menschliche Kognition sowie der Aufmerksamkeit, soll im Rahmen dieser Ausarbeitung zu einer Sensibilisierung bei der Konzeption von angemessenen Gestaltungslösungen von

Ereignis-Benachrichtigungen führen und wird in Kapitel 4 dieser Ausarbeitung behandelt.

2.1.2 Awareness

Der Begriff der Awareness lässt sich nur schwer eindeutig definieren und kann je nach Kontext eine unterschiedliche Bedeutung beinhalten. Im Kontext verteilter Systeme, besteht ein Ziel des Forschungsgebietes der Awareness darin, den Nutzern eines verteiltes Systems ein Bewusstsein über den Gruppenkontext einer kollaborativen Situation zu vermitteln, ohne dass dieser seine Aufmerksamkeit bewusst auf die entsprechenden Informationen lenken muss. Da die Nutzer der verteilten Simulationsumgebung RCE insbesondere ein Bewusstsein über verteilt eintretende Ereignisse entwickeln können sollen, sieht diese Ausarbeitung eine angemessene Betrachtung des Awareness-Begriffs vor.

2.1.3 CSCW-Systeme

Aufgrund der verteilten Aspekte des RCE-Systems, fand auch eine Betrachtung verschiedener CSCW-Forschungsgebietes statt. Durch eine Recherche wissenschaftlicher Ausarbeitungen über CSCW-Systeme und die Vermittlung von Ereignissen in diesen, soll ein allgemeines Verständnis über besondere Anforderungen an Benachrichtigungen in RCE entwickelt werden. Um zu überprüfen, ob sich diese Anforderungen an CSCW-Systemen überhaupt auf RCE übertragen lassen, findet in der Analysephase daher zunächst ein Vergleich des RCE-Systems und dem Begriff der CSCW-Systeme statt.

2.1.4 Darstellungsformen für Benachrichtigungen

Aufgrund einer Vielzahl möglicher Darstellungen für Informationen in Informations-Systemen, findet in der Analysephase eine Betrachtung gängiger Darstellungsformen statt. In dem Forschungsgebiet der Mensch-Computer-Interaktion, stellt die effiziente Vermittlung von Informationen jeglicher Art eine besondere Herausforderung an das System dar. Diese Herausforderung führte zu der Entwicklung diverser Guidelines¹⁰ Best Practices oder Styleguides verschiedener Software-Systeme, in denen unter anderem beschrieben wird, wie Informationen effizient vermittelt werden können.

¹⁰ „Guidelines“ definieren, wie angemessene Gestaltungslösungen erstellt werden können

2.2 Konzeptphase

In dieser Phase erfolgt die Erstellung eines allgemeinen Konzeptes für ein Benachrichtigungs-Systems, sowie die Entwicklung von angemessenen Gestaltungslösungen, die die wesentlichen Aspekte des Konzeptes abbilden sollen. Das Konzept basiert auf den gewonnenen Erkenntnissen aus der Analysephase, sowie auf den Anforderungen aus einer zuvor durchgeführten Nutzungskontextanalyse¹¹. Aufgrund der Vielfalt der zu berücksichtigenden Domänen der Mensch-Computer-Interaktion, fokussiert die Erstellung des Konzeptes dabei jedoch nicht auf allen identifizierten Anforderungen des Anforderungskataloges, sondern beschränkt sich auf den als wesentlich Erachteten Anforderungen. In dieser Phase findet daher eine Konzeption folgender Bestandteile statt:

Tagging-System

Das Tagging-System ermöglicht es dem Nutzer, beliebige Artefakte des RCE-Systems zu markieren, um sich zu diesen individuelle Informationen anzeigen zu lassen.

Notification-Center

Das Notification-Center stellt das wesentliche Konzept des Benachrichtigungs-Systems dar und beinhaltet alle relevanten Ereignis-Benachrichtigungen.

Notification-Bar

Die Notification-Bar ermöglicht es dem Nutzer, ein Bewusstsein über eingehende Benachrichtigungen zu entwickeln. Es wird darüber hinaus eingesetzt, um das Notification-Center zu öffnen.

Individualisierbarkeit des Benachrichtigungs-Systems

Aufgrund der variierenden Anforderungen verschiedener Nutzer, sieht das Konzept des Benachrichtigungs-Systems angemessene Individualisierungs-Möglichkeiten vor.

Komponenten-Awareness

Damit der Nutzer ein besseres Bewusstsein über die Verfügbarkeit bzw.

¹¹ Szafranauer M. 2015

Funktionsfähigkeit verteilt verfügbarer Komponenten entwickeln kann, soll dieser hinreichende Informationen über die Komponenten-Zustände vermittelt bekommen.

Eine ausführliche Beschreibung der zugrundeliegenden Konzepte kann dem Kapitel 5 dieser Ausarbeitung entnommen werden.

2.3 Entwicklung von Gestaltungslösungen

Ein wesentlicher Bestandteil bei der Erstellung eines Konzeptes unter Verwendung nutzerzentrierter Ansätze besteht unter anderem darin, entsprechende Gestaltungslösungen aus den identifizierten Anforderungen zu generieren. Diese können die Entwickler eines Konzeptes dabei unterstützen, ihre Visionen und Gedanken nach außen zu tragen und diese so anderen Menschen zugänglich zu machen. So lassen sich bereits in frühen Phasen der Konzeption kritische Design-Schwächen identifizieren und iterativ überarbeiten. Die Erstellung von Gestaltungslösungen soll daher allgemein nicht in einem einzelnen, klar abgrenzbaren Prozess erfolgen. Eher empfiehlt es sich, erste Entwürfe¹² in mehreren iterativen Entwurfsphasen bereits zu Beginn einer Konzeption zu erstellen. Durch dieses Vorgehen wird unter anderem gewährleistet, dass erste Vorstellungen bzw. Visionen festgehalten werden können, um diese im weiteren Verlauf der Konzeptionierung berücksichtigen zu können.

Bei der Entwicklung von Gestaltungslösungen, wurden verschiedene Methoden des Prototypings eingesetzt. Während erste Visionen zu Beginn der Konzeptphase durch Papier-basierte Mockups festgehalten wurden, erfolgte mit der fortschreitenden Entwicklung des Konzeptes die Erstellung von allgemeinen Wireframe-Modellen und detaillierten Mockups einzelner Konzept-Teile. Für eine einfachere iterative Überarbeitung bereits erstellter Gestaltungslösungen, wurden zu einem späteren Zeitpunkt primär digitale Mockups, durch den Einsatz angemessener Mockup-Tools wie Balsamiq¹³ erstellt und je nach Anforderungen angepasst. Bei der Erstellung der Gestaltungslösungen, wurden verschiedene Methoden nutzerzentrierter Ansätze berücksichtigt, um ein hohes Maß an Gebrauchstauglichkeit einhalten zu können. So wurden unter anderem die Grundsätze der Dialoggestaltung nach der DIN EN ISO 9241 Teil 110 berücksichtigt.

Um die möglichen Handlungsabläufe potenzieller Nutzer des Benachrichtigungs-Systems besser nachvollziehen zu können, und um eine Grundlage für den Einsatz angemessener Evaluations-Methoden zu bilden, erfolgte zudem die Erstellung eines pseudo-funktionalen Click-Dummies, in dem die Interaktionsmöglichkeiten mit dem

¹² Wie z.B. Paper-based Mockups

¹³ <https://balsamiq.com/> (Letzter Zugriff: Februar 2015)

Konzept des Notification-Centers simuliert werden. Die Erstellung dieses Prototypens erfolgte unter anderem auf Basis von Screenshots des bisherigen RCE-Systems und einer Erweiterung um das Benachrichtigungs-System als zusätzliche Komponente. Für die Entwicklung dieses Prototypens wurde das Mockup-Tool Axure¹⁴ in der Version 7.0 eingesetzt (siehe Abb. 1.).

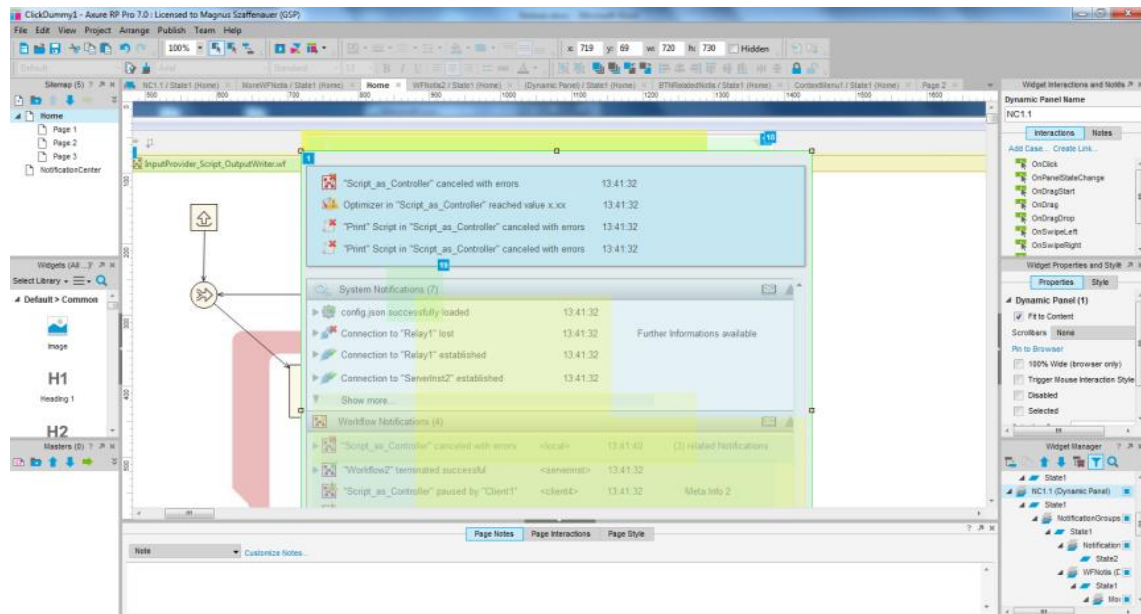


Abbildung 1. Editor des Axure Mockup-Tools mit einem Click-Dummy des Notification-Centers

2.4 Evaluation der Gestaltungslösungen

Gemäß der in Kapitel 1.2. erläuterten Problemstellung, bestand ein Schwerpunkt dieser Ausarbeitung unter anderem darin, geeignete Methoden der Evaluation für das erstellte Konzept zu identifizieren, diese vorzubereiten und anschließend durchzuführen. Um Evaluations-Methoden zu identifizieren, die sich im Kontext dieses Projektes anbieten, bestand die Methodologie in dieser Phase zunächst darin, die für eine Evaluation zur Verfügung stehenden Ressourcen zu ermitteln. Dies ist als Vorbereitung an eine Evaluation sinnvoll, um aus der Vielzahl an Evaluations-Methoden angemessene Methoden auszuwählen und ein geeignetes Vorgehensmodell aus diesen entwickeln zu können. So erfolgte unter anderem eine Betrachtung der zur Verfügung stehenden Gestaltungslösungen, der Zeit und der potenziellen Nutzer als Ressourcen der Evaluation.

Bevor in der Phase der Evaluation eine ausführlichere Abwägung verschiedener Evaluations-Methoden auf Basis der identifizierten Ressourcen erfolgen konnte, mussten darüber hinaus zunächst angemessene Metriken definiert werden,

¹⁴ <http://www.axure.com/> (Letzter Zugriff: März 2015)

die bei der Durchführung der Evaluation erfasst werden sollen. Die zu erfassenden Metriken können sich dabei auf die Wahl der zur Verfügung stehenden Evaluations-Methoden auswirken, wenn sich bestimmte Metriken beispielsweise nur durch eine bestimmte Methode erheben lassen. Die Festlegung der als sinnvoll erachteten Metriken basiert dabei unter anderem aus den gewonnenen Erkenntnissen der Analysephase, wie beispielsweise den „utility benefits“ aus dem Modell der Aufmerksamkeits-Kosten und Notification-Nutzen nach McCrickard et.al.¹⁵.

Anhand der identifizierten Metriken ließen sich für eine effiziente Evaluation anschließend geeignete Fragestellungen für Interviews generieren oder verschiedene Walkthrough-Methoden der Evaluation durchführen.

Der Prozess der Evaluation in dieser Ausarbeitung sieht insgesamt zwei grundlegende Ansätze an Evaluations-Methoden vor, die zu einer ganzheitlichen Evaluation aller relevanten Faktoren für gebrauchstaugliche Ereignis-Benachrichtigungen führen sollen:

- Analytische Evaluation
- Empirische Evaluation

Aufgrund der nur bedingt zur Verfügung stehenden Ressourcen, wird der Fokus in dieser Arbeit auf die Durchführung einer analytischen Evaluation auf Grundlage verschiedener Walkthrough-Methoden wie dem Cognitive Walkthrough, dem Pluralistic Walkthrough und dem Groupware-Walkthrough gelegt. Für die empirische Evaluation des Konzeptes hingegen, wird ein allgemeines Vorgehen vorgeschlagen, welches durchgeführt werden sollte, um eine ganzheitliche Identifikation weiterer potenzieller Design-Schwächen des erstellten Konzeptes zu ermöglichen. Da die zur Verfügung stehenden Ressourcen für eine solche empirische Evaluation im Kontext dieses Projektes unzureichend waren, ist dieser Teil der Evaluation als möglicher Ausblick einer weiterführenden Arbeit an dem erstellten Konzept zu betrachten. Die Planung sowie die Durchführung der Evaluation lässt sich Kapitel 7 dieser Ausarbeitung entnehmen.

2.5 Iterative Überarbeitung

Mit der Durchführung dieses Projektes, fand eine stetige iterative Überarbeitung verschiedener Gestaltungslösungen statt, die sowohl auf Basis einer Absprache mit den Entwicklern des RCE-Teams, als auch auf den Erkenntnissen der abschließenden Evaluation erfolgte. Es ist daher im Rahmen dieser Ausarbeitung schwierig, eine eindeutige Beschreibung der iterativen Prozesse vorzunehmen. Die allgemeine

¹⁵ McCrickard D. 2003, S.70

Methodologie während der Durchführung sieht demnach keine eindeutig abgrenzbaren Iterationszyklen vor. Vielmehr, wurde der Fokus auf viele kleinere iterative Überarbeitungen gelegt, als sich allein auf eine abschließende iterative Überarbeitung auf Basis einer umfangreichen Evaluation gegen Ende der Konzeptphase zu stützen. Allgemein orientiert sich das Vorgehen dabei an dem „Prozess zur Gestaltung gebrauchstauglicher interaktiver Systeme¹⁶“, in dem ein vermehrt iterativer Prozess beschrieben wird.

¹⁶ DIN EN ISO 9241 Teil 210 (2010)

3 Remote Component Environment

In diesem Kapitel findet eine Betrachtung der verteilten Simulationsumgebung RCE (Remote Component Environment) statt, die unter anderem durch die Einrichtung der „Simulations- und Softwaretechnik“ des DLR entwickelt und gemäß der wachsenden Anforderungen der Nutzer sowie einer proaktiven Entwicklung neuer Funktionalitäten stetig erweitert wird. Der Fokus dieses Kapitels liegt dabei nicht auf einer ganzheitlichen Definition der Systemarchitektur von RCE und dessen Funktionalitäten, sondern soll dem Leser die Anwendungsgebiete des Systems, sowie die grundlegenden relevanten Begriffe innerhalb der Domäne nahelegen, damit entsprechende Begrifflichkeiten im weiteren Verlauf dieser Ausarbeitung verwendet werden können. Da dieses Projekt die Konzeptionierung eines gebrauchstauglichen Benachrichtigungs-Systems für RCE vorsieht, wird darüber hinaus ausführlicher auf die in RCE auftretenden Ereignisse eingegangen und die Anforderungen an eine gebrauchstaugliche Darstellung dieser, gemäß einer zuvor durchgeführten Nutzungskontextanalyse verdeutlicht.

3.1 Das System

Das RCE-System lässt sich unter anderem anhand der entsprechenden Definition der Webpräsenz allgemein beschreiben:

„RCE is a distributed, workflow-driven integration environment. We made RCE especially for engineers and scientists to design, analyze, and optimize complex systems (e.g., aircrafts, ships, or satellites) by using and integrating their own design and analysis tools.“¹⁷

Demnach beschreibt RCE eine Workflow-gesteuerte verteilte Simulationsumgebung, die es Ingenieuren und Wissenschaftlern ermöglicht, komplexe Systeme (wie Flugzeuge, Schiffe oder Satelliten) zu entwerfen, zu analysieren und zu optimieren. Die Nutzer des Systems können dabei ihre eigenen Entwurfs- sowie Analyse-Werkzeuge (die so genannten Tools) in die Simulationsumgebung integrieren, um so eine gegebene Problemstellung zu bearbeiten. Um dies zu ermöglichen, greift RCE auf den Einsatz der nachfolgenden Komponenten zurück

¹⁷ www.rcenvironment.de (Letzter Zugriff: April 2015)

3.1.1 Workflows

Das Konzept der Workflows stellt in der verteilten Simulationsumgebung RCE das grundlegende Artefakt dar, anhand derer sich komplexe Systeme analysieren und entwickeln lassen. Die in RCE durch einen Nutzer erstellten Workflows, bestehen dabei aus verschiedenen lokal sowie verteilt verfügbaren Tools, die verschiedene Funktionalitäten (wie z.B. Simulationen oder komplexe Berechnungen) realisieren. Durch den Einsatz dieser Tools, lässt sich so interdisziplinäres Wissen in einem Workflow zusammenführen, um beispielsweise eine komplexe Problemstellung zu bewältigen. Damit sich RCE in einer Vielzahl an Domänen einsetzen lässt, können die in den Workflows eingesetzten Tools durch die Nutzer eigenständig in die Simulationsumgebung eingebunden werden. Der Nutzer hat so die Möglichkeit, seine Expertise durch die Integration eines eigenen Tools durch das RCE-System bereit zu stellen. Die in einem Workflow enthaltenen Tools, können über verschiedene Ein- bzw. Ausgänge dieser entsprechend einer gegebenen Problemstellung verknüpft werden.

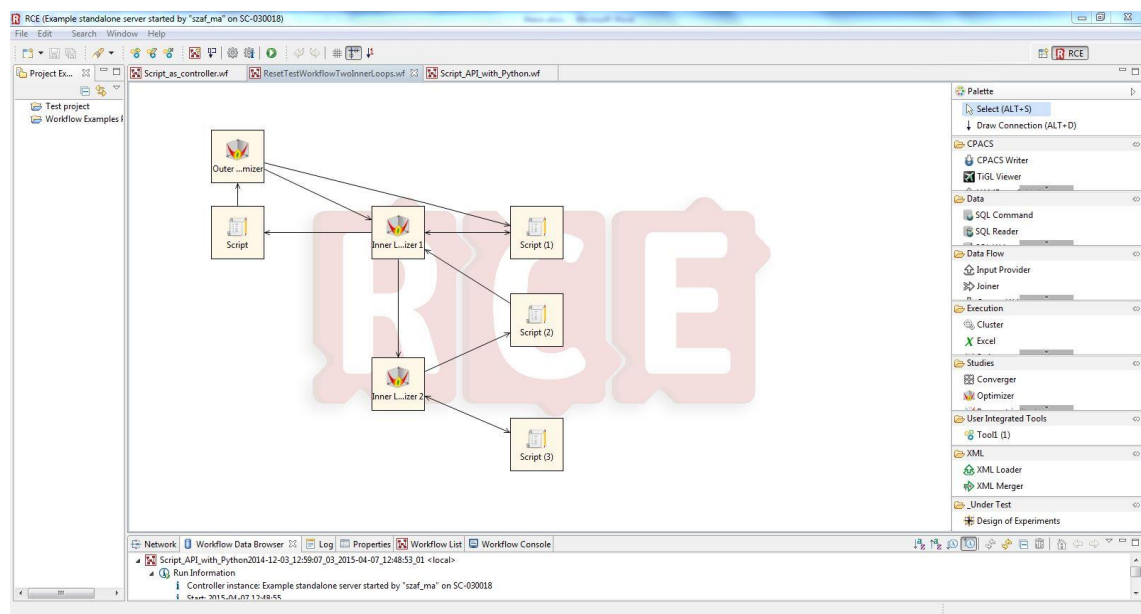


Abbildung 2: Darstellung eines Workflows in RCE

3.1.2 Tools

Die Tools in RCE beschreiben die wesentlichen Bestandteile eines Workflows und werden über verschiedene Quellen des RCE- Systems bereitgestellt. Neben nativ enthaltenen Tools in RCE, die nach einer Installation des System bereits verfügbar sind,

kann der Nutzer eigene Tools integrieren, oder auf verteilt verfügbare Tools zurückgreifen. Während in den Tools der Nutzer, spezifische Funktionalitäten gemäß der Anforderungen eines Projektes enthalten sind, beschreiben die in RCE nativ enthaltenen Tools primär Hilfs-Funktionalitäten, wie eine Script-Komponente, einen „Optimizer“ oder eine „Parametric Study“-Komponente. Diese Komponenten realisieren elementare Funktionalitäten, die in einer Vielzahl an Workflows erforderlich sind und daher mit der Weiterentwicklung von RCE in die Umgebung integriert wurden. Im weiteren Verlauf dieser Arbeit, soll der Begriff der „Tools“ allgemein für alle in RCE verfügbaren Tools eingesetzt werden, während der Begriff Komponente sich lediglich auf die in RCE nativ enthaltenen Tools bezieht.

Je nach Anforderungen an die, in einem Workflow eingesetzten, Komponenten können diese eine beliebige Anzahl an Ein- sowie Ausgängen besitzen. Damit eine Komponente innerhalb eines Workflows durchgeführt werden kann, müssen an den definierten Eingängen dieser Komponente zunächst Daten anliegen.

3.1.3 Instanzen in RCE

Um dem Aspekt der Verteiltheit gerecht werden zu können, lassen sich mehrere RCE-Instanzen in einer Netzwerkkumgebung einsetzen. Die Einsatzgebiete dieser RCE-Instanzen reichen von Client-Instanzen, in denen die Nutzer eigene Workflows entwickeln oder eigene Tools integrieren, über „Tool-Server“ bis hin zu „Compute-Nodes“. Grundsätzlich kann dabei jede RCE-Instanz in einer Netzwerkkumgebung verschiedenen Einsatzgebieten zugeordnet werden. So lässt sich eine RCE-Instanz sowohl als Client zur Erstellung von Tools, als auch als Tool-Server einsetzen, um entwickelte Tools anderen Nutzern zur Verfügung zu stellen.

Zwar erfordern diese Einsatzgebiete einer RCE-Instanz keine Anpassung des grundlegenden Systems, jedoch kann das RCE-System durch eine entsprechende Konfiguration auf ein bestimmtes Einsatzgebiet angepasst werden. Während eine RCE-Instanz als Compute-Nodes eingesetzt wird, um besonders rechenintensive Workflows oder Tools zu berechnen, ohne die Client-Instanz eines Nutzers unnötig zu belasten, dient die Verwendung eines Tool-Servers dazu, entwickelte sowie integrierte Tools anderen Nutzern zur Verfügung zu stellen. Die in RCE verfügbaren Tools können darüber hinaus ebenfalls instanziiert werden. So ist es unter anderem möglich, dass ein Tool in einer RCE-Instanz mehrfach instanziiert wird, indem dieses z.B. in mehreren Workflows enthalten ist oder in einem Workflow mehrfach enthalten ist. Die Verwendung von RCE-Instanzen zur Durchführung von Workflows oder Tools, kann durch den Nutzer beliebig angepasst werden. So kann der Nutzer unter anderem selbst definieren, auf welcher RCE-Instanz ein Workflow oder die darin enthaltenen

Komponenten berechnet werden. Ein grundlegendes Verständnis über die möglichen Instanziierungen in RCE ist insbesondere bei der Konzeption eines gebrauchstauglichen Benachrichtigungs-Systems notwendig, um die dem Nutzer relevanten Informationen, wie die entsprechenden Ereignis-Instanzen, effizient zu vermitteln.

3.2 Darstellung von Ereignissen in RCE

Für die Konzeption eines gebrauchstauglichen Benachrichtigungs-Systems der Ereignisse in RCE, ist es notwendig den Ist-Zustand des Systems zu erfassen und die relevanten Ereignisse zu identifizieren. Die Analyse der verschiedenen Darstellungsformen für Ereignisse in RCE erfolgte dabei unter anderem im Rahmen einer Nutzungskontextanalyse. Folgende Bestandteile der RCE-Benutzeroberfläche ließen sich dabei als Informationsquellen für verschiedene Informationen erheben.

3.2.1 Network-View

Über die Network-View kann der Nutzer neue Verbindungen zu RCE-Instanzen aus seiner Netzwerkumgebung hinzufügen. Darüber hinaus zeigt diese Ansicht an, zu welchen Netzwerken er aktuell verbunden ist und welche RCE-Instanzen in seiner Umgebung aktuell aktiv sind. Der unerwartete Abbruch einer Verbindung wird dem Nutzer durch einen Popup-Dialog, sowie die entsprechende Verbindung in der Network-View als „disconnected“ dargestellt.

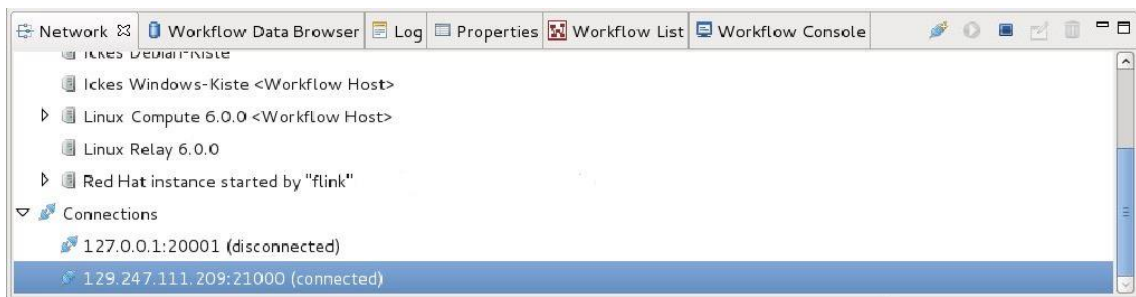


Abbildung 3: Network-View von RCE mit Darstellung der RCE-Instanzen

3.2.2 Workflow Data Browser-View

Der „Workflow Data Browser“ ermöglicht es dem Nutzer, ausführliche Informationen über die Daten seiner Workflows zu erhalten. Neben den eingehenden und ausgehenden Werten der verschiedenen Komponenten eines Workflows, erhält der Nutzer auch

Informationen über die Laufzeit eines Workflows, sowie dessen aktuellen Zustand. Um zu erfahren, ob ein Workflow noch durchgeführt wird oder dieser bereits terminiert ist, kann der Nutzer den Status seiner Workflows über eine „Refresh“-Funktionalität aktualisieren.

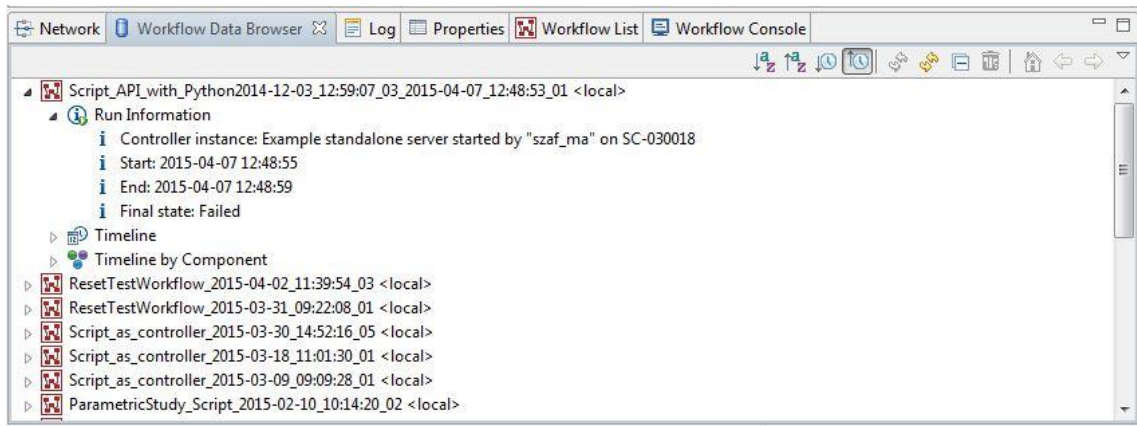


Abbildung 4: Workflow Data Browser des RCE-Systems

3.2.3 Log-View

In der Log-View erhält der Nutzer Informationen über eintretende Fehler-Ereignisse oder Warnhinweise des RCE-Systems. Die Art der Ereignisse wird dabei über grafische Indikatoren verdeutlicht (siehe Abb. 5) und eindeutig einer Instanz zugeordnet. Zudem enthält diese Darstellung auch Informationen über den Zeitpunkt, der eingetretenen Meldungen. Während in der tabellarischen Ansicht nur ein kurzer Informationstext zu einer Meldung angezeigt wird, kann sich der Nutzer durch eine Auswahl der gewünschten Meldung weitere Informationen anzeigen lassen.

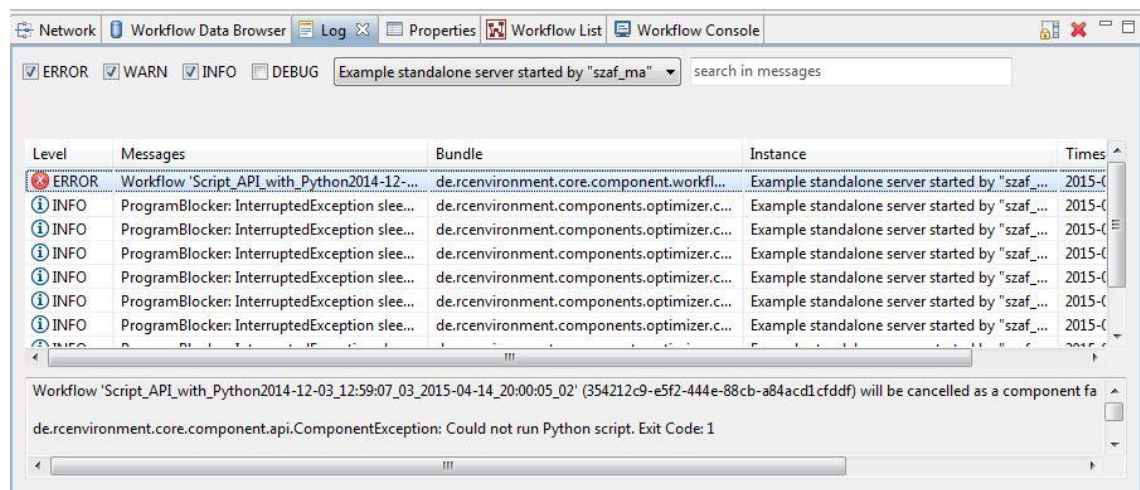


Abbildung 5: Log-View in RCE mit Darstellung der grafischen Indikatoren

3.2.4 Workflow List-View

In der Workflow List-View, werden dem Nutzer grundlegende Informationen über alle sichtbaren Workflows in der verteilten Simulationsumgebung angezeigt. Neben Workflows, die der Nutzer selber durchführt, erfolgt auch eine Darstellung der Workflows anderer Nutzer, sofern diese als „Workflow-Host“ konfiguriert sind. Der Fokus dieser Darstellung liegt primär darauf, die einzelnen Zustände der Workflows den Nutzern zugänglich zu machen. Darüber hinaus kann der Nutzer anhand dieser Darstellung seine Workflows pausieren, abbrechen oder wieder fortsetzen.

Name	Status	Controller	Start	Started From	Additional Information
ParametricStudy_Script_...	Finished	Example standalone ser...	2015.04.14 20:04:34	Example standalone ser...	
ResetTestWorkflow_2015...	Running	Example standalone ser...	2015.04.14 19:59:58	Example standalone ser...	
Script_as_controller_2015...	Failed	Example standalone ser...	2015.04.14 20:04:19	Example standalone ser...	

Abbildung 6: Workflow List-View in RCE

3.2.5 Workflow Console-View

In der Workflow Console-View, erfolgt eine Darstellung aller Benachrichtigungen, die durch die Komponenten eines Workflows generiert werden. Neben einfachen „Console-

Outputs“ die in den jeweiligen Komponenten deklariert sind und eine Ausgabe realisieren, erfolgt hierbei auch eine Darstellung über die Zustands-Änderung einzelner Komponenten oder Fehlermeldungen dieser. Ähnlich der Log-View, werden die in der Workflow Console dargestellten Informationen durch grafische Indikatoren klassifiziert (siehe Abb. 7.).

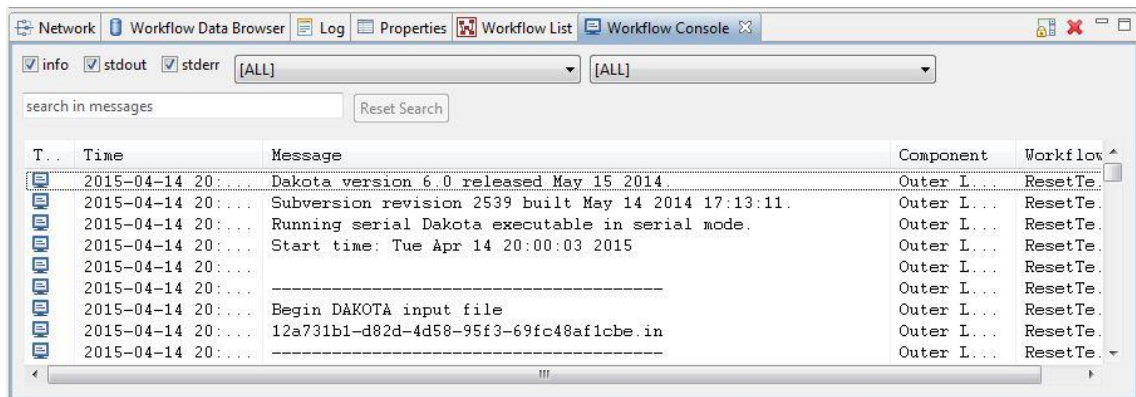


Abbildung 7: Workflow Console-View in RCE

3.2. Ereignisse in RCE

Auf Grundlage einer Betrachtung der vorhandenen Informationsquellen in RCE, sowie den Erkenntnissen aus der zuvor durchgeführten Nutzungskontextanalyse, kann anschließend eine Betrachtung der verschiedenen Ereignisse in RCE erfolgen, die bei der Konzeption eines gebrauchstauglichen Benachrichtigungs-System zu berücksichtigen sind. Viele der dabei als relevant erachteten Ereignisse basieren auf den in der Nutzungskontextanalyse identifizierten Anforderungen. So führt unter anderem die Anforderung

„1.1.1.7. Tool-Verfügbarkeit anzeigen“

zu den Ereignissen „Tool verfügbar“ bzw. „Tool nicht verfügbar“ oder die Anforderung

„1.1.1.9. Verteilten Tool-Einsatz anzeigen“

zu dem Ereignis „Tool wird verteilt eingesetzt“. Daraus ergeben sich für das RCE-System folgende Ereignisse, die für ein besseres Verständnis im Rahmen dieser Ausarbeitung in die Kategorien der Tool-Ereignisse, Workflow-Ereignisse und System-Ereignisse unterteilt wurden.

3.2.1. Tool-Ereignisse

Zu den Tool-Ereignissen zählen alle Ereignisse, die sich unmittelbar auf Änderungen der Verfügbarkeit oder des allgemeinen Status von Tools der Nutzer auswirken. Hierzu zählen:

Fehlerhaftes Tool

Ein eingesetztes Tool hat während seiner Durchführung in einem Workflow einen Fehler verursacht.

Tool-Anfrage

Ein Nutzer sendet eine Anfrage an ein benötigtes Tool an den Tool-Integrator oder den Administrator eines Tool-Servers. Für dieses Ereignis muss das Tool dem entsprechenden Nutzer als „nicht verfügbar“ angezeigt werden.

Tool verfügbar

Ein verteiltes Tool ist verfügbar und kann durch den Nutzer eingesetzt werden.

Tool nicht mehr verfügbar

Ein verteiltes Tool ist nicht mehr verfügbar und kann daher nicht mehr durch den Nutzer in seinen eigenen Workflow integriert werden. Bereits laufende Workflows in denen das Tool enthalten ist, können ggf. nicht erfolgreich terminieren.

Tool aktualisiert

Ein bereits verteilt verfügbares Tool wurde aktualisiert und besitzt daher ggf. abweichende Funktionalitäten, die vermittelt werden müssen.

3.2.2. Workflow-Ereignisse

Die Workflow-Ereignisse stellen grundlegende Informationen über die Durchführung von Workflows in der verteilten Simulationsumgebung RCE bereit. Folgende Workflow-Ereignisse sind dabei zu berücksichtigen:

Workflow gestartet

Ein erstellter Workflow wird mit einer bestimmten Instanz (Workflow-Host, Controller, etc) gestartet.

Workflow durchgeführt

Ein gestarteter Workflow wurde erfolgreich ohne Fehlermeldungen durch eine bestimmte Instanz durchgeführt.

Workflow mit Fehler abgebrochen

Ein Workflow in Durchführung konnte durch einen unerwarteten Fehler nicht erfolgreich terminieren.

Workflow pausiert

Ein Workflow in Durchführung wird durch eine bestimmte Instanz oder das System pausiert.

Workflow Zwischenergebnis darstellen

Das System stellt dem Nutzer Informationen über die für ihn relevanten Parameter während der Durchführung eines Workflows zur Verfügung.

Workflow Durchführungsdauer darstellen

Das System stellt dem Nutzer Informationen über die Durchführungsdauer eines laufenden Workflows zur Verfügung.

3.2.6 System-Ereignisse

Zu den Systemereignissen zählen alle Ereignisse, die durch das System ausgelöst wurden oder eine Änderung des System-Zustandes bewirken.

Netzwerkverbindung hergestellt

Das System hat die Verbindung zu einer, durch den Nutzer eingetragenen IP-Adresse erfolgreich nach Start einer RCE-Instanz oder dem Hinzufügen der entsprechenden Verbindung hergestellt.

Netzwerkverbindung unterbrochen

Die Netzwerkverbindung zu einer der durch den Nutzer eingetragenen RCE-Instanz wurde unterbrochen

Systemfehler

Während der Durchführung von Workflows oder der allgemeinen Interaktion mit dem RCE-System tritt ein unerwarteter Systemfehler auf.

Unzureichender Speicherplatz

Während der Entwicklung von Workflows oder der Durchführung dieser, reicht der verfügbare Speicherplatz nicht mehr für eine weitere Verarbeitung der Daten aus.

Externer Einfluss

Ein externer Einfluss wirkt sich auf das System aus und kann zu einer Änderung des System-Status führen. Die externen Einflüsse können unter Umständen nur schwer durch das System erfasst und deren Ursprung über eine Ereignis-Benachrichtigung vermittelt werden. Hier kann exemplarisch das unabsichtliche Löschen der RCE-Konfigurationsdatei genannt werden oder eine mangelnde Netzwerkkonnektivität aufgrund einer Auslastung des Netzwerkes durch andere Systeme

Die soeben dargestellten Ereignisse, werden für eine effiziente Identifikation der Anforderungen an entsprechende Ereignis-Benachrichtigungen in Kapitel 4 ausführlicher behandelt, indem diese in eine, dem Projekt-Kontext angemessene Klassifizierung überführt werden. An dieser Stelle sei zudem anzumerken, dass im Rahmen dieser Ausarbeitung und einer Erstellung eines gebrauchstauglichen Konzeptes nicht alle identifizierten Ereignisse gleichermaßen berücksichtigt wurden. Eine Abwägung der als kritisch erachteten Ereignisse erfolgte unter anderem durch eine Gewichtung der zugrunde liegenden Anforderungen im Prozess der Nutzungskontextanalyse, sowie auf den, während der Analysephase gewonnenen Erkenntnissen (siehe Kapitel 4).

4 Analyse relevanter Konzepte und Begrifflichkeiten

In diesem Kapitel folgt eine Betrachtung relevanter Begrifflichkeiten und Konzepte, die für eine qualitativ hochwertige und ganzheitliche Konzeption eines Benachrichtigungs-Systems in einer verteilten Simulationsumgebung berücksichtigt wurden. Um den in Kapitel 1.2. vorgestellten Forschungsfragen gerecht zu werden, erfolgte zunächst eine Analyse verschiedener wissenschaftlicher Arbeiten, die sich in den Domänen der CSCW-Systeme und der Vermittlung von Benachrichtigungen in diesen auseinandersetzen. Da die Forschungsgebiete des soeben vorgestellten Kontextes vielseitig und nicht eindeutig zu benennen sind, fällt es mitunter schwer, eine umfangreiche Analyse aller Konzepte vorzunehmen, die in Relation zu dieser Arbeit stehen. Aus diesem Grund befasst sich dieses Kapitel primär mit den Begrifflichkeiten und Konzepten, die sich unmittelbar auf die Konzeption des Benachrichtigungs-Systems auswirken und daher als wichtig betrachtet werden. Auf Basis der Begriffsdefinition wird anschließend ein Modell vorgestellt, mit dessen Hilfe die Anforderungen an Benachrichtigungen gemäß verschiedener Ereignisse identifiziert werden können.

4.1. Relevante Konzepte

Im Rahmen dieser Ausarbeitung werden verschiedene grundlegende Begrifflichkeiten verwendet, die je nach Kontext eine stark abweichende Bedeutung beinhalten können. Aus diesem Grund erfolgen in diesem Abschnitt eine Analyse relevanter Begriffe, sowie eine Betrachtung derer zugrundeliegender Konzepte, um deren Relevanz für diese Ausarbeitung zu verdeutlichen. Die hierbei vorgestellten Begriffe und Konzepte können nach einer ausführlichen Diskussion anschließend in die Konzeption eines geeigneten Notifications-Systems berücksichtigt werden.

4.1.1 CSCW-Systeme

Das Forschungsgebiet der CSCW-Systeme¹⁸ befasst sich mit der Zusammenarbeit zwischen Menschen in Gruppensituationen, unter der Verwendung rechnergestützter Informations- bzw. Kommunikationstechniken. CSCW fokussiert dabei primär auf die Kooperation zwischen einzelnen Teilnehmern einer Gruppe und versucht unter anderem, diese durch angemessene Systeme physisch voneinander zu entkoppeln. Bei CSCW handelt es sich um ein interdisziplinäres Forschungsgebiet, welches sich neben

¹⁸ Computer-supported-collaborative-work

der Domäne der Informatik auch mit Sozialwissenschaften, Organisationswissenschaften, Arbeitswissenschaften und vielen weiteren befasst¹⁹. Diese Tatsache erschwert es, umfangreiche Grundlagen oder eine allgemeine Begriffsdefinition über die Thematik zu vermitteln, weshalb von einer ausführlichen Definition sowie einer Vermittlung von Grundlagen im Rahmen dieser Arbeit abgesehen wird. Interessant ist das Forschungsgebiet der CSCW-Systeme jedoch insbesondere aus Sicht der Informatik und speziell der Mensch-Computer-Interaktion, weshalb die wesentlichen Aspekte eines CSCW-Systems in diesem Kontext erläutert werden sollen. Die dabei gewonnenen Erkenntnisse können anschließend bei einer Konzeption möglicher Darstellungsformen für eine verteilte Simulationsumgebung wie RCE berücksichtigt werden.

Der Vorteil bei einer kooperativen Aufgabenbewältigung besteht insbesondere darin, dass sich die einzelnen Mitglieder einer Gruppe mit deren individuellen Fähigkeiten sowie Fertigkeiten einander ergänzen können, um so gegebene Ziele effizienter zu erreichen. Diesen Vorteil auf ein computergestütztes System zu übertragen, ist bei genauerer Betrachtung aber nicht trivial zu lösen und erfordert eine intensive Auseinandersetzung mit dem jeweiligen Gruppenkontext.

So steht die Software-Ergonomie als Disziplin der Mensch-Computer-Interaktion im Kontext von CSCW-Systemen neuen Herausforderungen gegenüber, wenn eine kooperative Zusammenarbeit und die Entwicklung eines Gruppen-Bewusstseins gewährleistet werden soll, ohne dabei den Mensch bei der Aufgabenbewältigung negativ zu beeinträchtigen. Es empfiehlt sich daher für die Entwickler eines CSCW-Systems, die Position eines „Ethnographen“ einzunehmen und sich in die bestehende Gruppe für eine ganzheitliche Untersuchung zu integrieren. So kann dieser alle relevanten Artefakte identifizieren, die im jeweiligen Kontext für eine kollaborative Aufgabenbewältigung notwendig sind und diese auf computergestützte Informations- und Kommunikationstechniken übertragen. Da die Vermittlung eines Gruppen-Kontextes über ein computergestütztes System nicht ohne einen besteht eine der Schwerpunkte darin, das Senden sowie das Empfangen von Gruppen-relevanten Informationen unter Berücksichtigung gebrauchstauglicher Aspekte zu realisieren.

Ein weiterer wichtiger Aspekt aus Sicht der Informatik-Domäne betrifft die Sicherheit bei einer Kooperation über ein CSCW-System. Gerade bei verteilten Systemen, bei der durch den Einsatz verschiedener technischer Kommunikationsmittel eine verteilte Kooperation realisiert wird, treten Risiken auf, die in jedem Fall zu berücksichtigen sind.

¹⁹ Mills K. 2003

Möchte man nun sicherheitsrelevante Aspekte in einer verteilten Simulationsumgebung realisieren, so steht dies oftmals in einem Konflikt mit der Gebrauchstauglichkeit bzw. Software-Ergonomie eines Systems. Durch den Einsatz verschiedener Sicherheits-Mechanismen muss der Nutzer eines Systems in der Regel Einschränkungen in der Gebrauchstauglichkeit akzeptieren. Hierbei ist eine angemessene Abwägung aus Sicherheits-Aspekten und der Gebrauchstauglichkeit des jeweiligen Systems vorzunehmen. Bei einer ausführlichen Auseinandersetzung wird zudem deutlich, dass sich hierbei auch Aspekte der Sozialwissenschaften widerspiegeln. So ist die Akzeptanz, ein verteiltes System für kollaboratives Arbeiten einzusetzen, oftmals noch gering, da die potenziellen Nutzer solchen Systemen skeptisch gegenüber stehen. Besonders kritisch ist dies bei CSCW-Systemen, die den Einsatz von audiovisuellen Peripherie-Geräten, wie Webcams oder Mikrofonen vorsehen. So besteht unter anderem die „Angst“, dass solche Systeme durch Unbefugte abgehört werden, was zu einer Verletzung der Privatsphäre führen kann.

Natürlich sind dies nur einige Punkte die im Bezug zur Mensch-Computer-Interaktion berücksichtigt werden sollten. Im Kontext dieses Projektes stellt sich zudem die Frage, in wie weit sich die verteilte Simulationsumgebung RCE dem Begriff der CSCW-Systeme zuordnen lässt. Bei einer ausführlicheren Betrachtung wird deutlich, dass das RCE-System zwar nicht als eindeutig den klassischen CSCW-Systemen zugeordnet werden kann, jedoch einige Aspekte mit diesen gemeinsam hat. So findet auch in RCE, durch den Einsatz eigens entwickelter Tools verschiedener Nutzer, eine kollaborative Aufgabenbewältigung statt, in der gemeinsam an verschiedenen Aufgabenstellungen gearbeitet wird. Im Gegensatz zu CSCW-Systemen, findet die kollaborative Aufgabenbewältigung hierbei jedoch eher indirekt statt. Während in CSCW-Systemen eine kollaborative Bearbeitung von Artefakten durch mehrere Nutzer überwiegend zeitgleich oder sehr direkt stattfindet, müssen die Nutzer des RCE-Systems bislang nur selten eine unmittelbare Kommunikation mit anderen Nutzern über das System eingehen. Dementsprechend stellt RCE zum Zeitpunkt dieser Ausarbeitung auch keine Mechanismen bereit, anhand derer sich andere Teilnehmer in einer Simulationsumgebung unmittelbar kontaktieren lassen. Sofern ein Austausch zwischen mehreren Nutzern erforderlich ist, müssen diese auf andere Methoden zurückgreifen (persönliches Treffen, telefonischer Kontakt, etc). Eine weitere Gemeinsamkeit, die sowohl in CSCW-Systemen als auch in RCE auftritt, ist die Vermittlung von kontextrelevanten Informationen, um den Nutzern des Systems ein Bewusstsein über die kollaborative Situation zu ermöglichen. So ist es für eine effiziente Aufgabenbewältigung in RCE unter anderem erforderlich, dass die Nutzer wissen welche verteilten Tools im aktuellen Kontext zur Verfügung stehen oder welche Workflows von welchem Nutzer durchgeführt werden. Damit RCE eindeutig dem

Begriff der CSCW-Systeme zugeordnet werden kann, muss dieses zunächst um weitere Mechanismen erweitert werden, die eine engere Kopplung zwischen Nutzern einer verteilten Simulationsumgebung realisieren. In wie fern dies für das RCE-System sinnvoll ist, sollte dabei jedoch zuvor identifiziert werden.

4.1.2 Aufmerksamkeit

Eine fundamentale Forschungsdomäne im Kontext von CSCW-Systemen befasst sich unter anderem mit der Aufmerksamkeit des Menschen. Die Aufmerksamkeit steuert dabei, welche äußeren Reize aus der Umwelt des Menschen für eine weitere Verarbeitung durch kognitive Prozesse gefiltert und weitergeleitet werden. Gemäß verschiedener Theorien²⁰ der Aufmerksamkeit, sieht sich der Mensch stets einer Fülle an Reizen konfrontiert. Die kognitive Verarbeitung eines Menschen stellt in diesen Theorien oftmals jedoch eine stark eingeschränkte Ressource dar, weshalb eine Verarbeitung dieser Reize nach vielen Modellen nicht parallel erfolgen kann. Eine ausführliche Auseinandersetzung mit den verschiedenen Theorien der Aufmerksamkeit soll in diesem Kapitel allerdings nicht erfolgen, da sich diese nicht ohne erheblichen Aufwand in einem Kapitel zusammenfassen lassen und wohl eine eigenständige Ausarbeitung erfordern würden. Dennoch werden die als wesentlich erachteten Konzepte bzw. Aspekte betrachtet, um die Relevanz des Aufmerksamkeits-Begriffs im Kontext einer Konzeption geeigneter Ereignis-Benachrichtigungen zu verdeutlichen.

4.1.2.1 Filtermodell nach Broadbent

Das Filtermodell nach Donald Broadbent wurde 1958 entwickelt und beschreibt den Prozess von der sensorischen Reiz-Aufnahme eines Menschen, über die Verarbeitung eines gefilterten Reizes bis hin zur Reiz-Antwort. Das Modell (siehe Abb. 8) sieht dabei nach Eingang verschiedenster Reize über Sensoren²¹ einen sensorischen Puffer vor, den so genannten „short term store“, in dem zunächst alle eingehenden Reize für einen kurzen Zeitraum gespeichert werden²². Auf diesem „Pool“ an Reizen kann anschließend der selektive Filter angewandt werden, der einen zu verarbeitenden Reiz für die kognitive Verarbeitung filtert. Der selektive Filter stellt dabei eine Art Flaschenhals dar, in dem nur eine geringe Menge an Reiz-Informationen durchströmen kann. Die Selektion der zu verarbeitenden Reize kann durch den Menschen bewusst gesteuert werden, weshalb der Begriff der Aufmerksamkeit auch oft in direktem Bezug zum Bewusstsein eines Menschen steht. Der gefilterte Reiz kann anschließend nach der

²⁰ Wie beispielsweise die Theorie der „frühen Selektion“ nach Broadbent (1958) oder die „single pool“ Theorie nach Kahneman (1973)

²¹ Z.B. visuell, akustisch, physisch

²² Man geht hierbei von einem „Bruchteil“ einer Sekunde aus

kognitiven Verarbeitung im „Limited Capacity Processor“ in den „Long-Term Store“ für einen längeren Zeitraum gespeichert oder über das „Output-System“ eine Reaktion auf den verarbeiteten Reiz generiert werden.

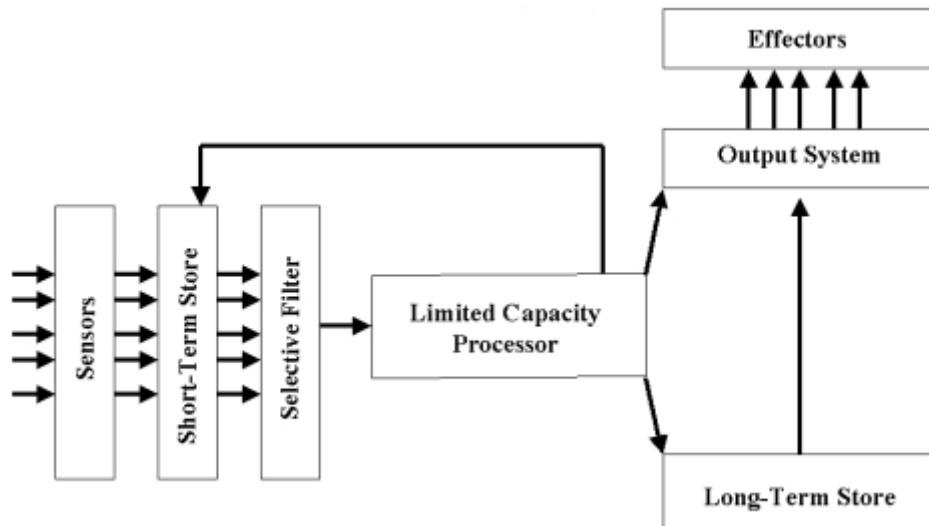


Abbildung 8: Modell der Aufmerksamkeit nach Broadbent

4.1.2.2 Aufmerksamkeit und Nutzen nach D. Scott McCrickard et.al.

Gemäß der Filtertheorie nach Broadbent und anderen Modellen auf den Forschungsgebieten der Aufmerksamkeit, stellt die Aufmerksamkeit sowie die kognitive Verarbeitung von eingehenden Reizen eine eingeschränkt verfügbare und somit sehr wertvolle Ressource dar. Steht nun die Konzeption geeigneter Darstellungsformen für eintretende Ereignisse in einer verteilten Simulationsumgebung wie RCE im Fokus, so stellt sich die Frage, in wie weit es vertretbar ist, die Aufmerksamkeit eines Nutzers weg von seiner eigentlichen Aufgabe, hin zu den verschiedenen Ereignissen zu lenken. Mit dieser Fragestellung befassten sich unter anderem auch D. Scott McCrickard und C.M. Chewar²³ und entwickelten ein Konzept, um das Verhältnis aus Aufmerksamkeit und den Nutzen bei der bewussten Aufmerksamkeits-Steuerung zu beschreiben. So definieren sie die größte Herausforderung bei der Darstellung von Notifications wie folgt:

²³ McCrickard D. 2003

„The paramount challenge of notification is preventing unwanted distraction to the primary task, while still delivering information in an accurate and timely manner. In many cases, very little distraction can be tolerated. For example, a typical in-vehicle information system may notify the user about navigation instructions, incoming communications, and other information secondary from the main task of the user—driving the car.”²⁴

Gemäß diesen Zitats, besteht die große Herausforderung darin, dem Nutzer relevante Benachrichtigungen möglichst zeitnahe zu vermitteln, ohne ihn dabei von seiner primären Aufgabe unnötig abzulenken. In vielen Fällen ist es jedoch tolerierbar, wenn ein Nutzer für einen kurzen Augenblick von seiner primären Aufgabe abgelenkt wird.

Um die Zusammenhänge aus Aufmerksamkeit und deren Nutzen für den Menschen zu verdeutlichen, stellen McCrickard und Chewar eine Tabelle bereit, in der diese näher erläutert werden. Gemäß dieser Tabelle erwarten die Nutzer beim Einsatz eines Notification-Systems gewisse Vorteile, die „utility benefits“, welche sich in insgesamt vier grundlegende Ziele klassifizieren lassen. Comprehension, Reaction und Interruption werden hierbei als kritische Parameter betrachtet, die sich im Rahmen des Design-Prozesses eines Notification-Systems als Metriken anwenden lassen und somit potenziellen Optimierungsbedarf adressieren. Um Vorteile aus dem Nutzen eines Notification-Systems zu ziehen, sind die Nutzer bereit, Kosten in Form von Aufmerksamkeit aufzubringen. Diese Kosten lassen sich der rechten Hälfte der Tabelle entnehmen. Die entstandenen Kosten als Faktor definieren dabei die Menge an Aufmerksamkeit, die von den primären Aktivitäten/Aufgaben eines Menschen bewusst auf Notifications „umgelenkt“ wird. Die in der Tabelle aufgeführten Kostenfaktoren bezeichnen dabei besondere Aspekte, die die Kosten der Aufmerksamkeit zusätzlich erhöhen können.

²⁴ McCrickard D. 2003, S.67

utility benefits		attention costs	
user goal	general goals	situation parameter	cost factors
identify state changes understand patterns and trends assimilate complex information monitor resources over time gain awareness of collaborators	Comprehension information is related to existing knowledge and stored for future use	Context	goal relationships of tasks task perceptual-motor qualities data-link dependencies relative tasks priorities interruptability focus/peripheral location platforms and environment
make decisions modify primary task approach provide response acknowledge status	Reaction immediate response to a notification stimulus, with or without shifting attention	User characteristics	skill and automaticity cognitive and perceptual abilities current overall mental workload sender/receiver roles demographics
pace daily activities prompt task transition receive urgent/timely information synchronize with colleagues	Interruption intentional and inherently useful reallocation of attention from other tasks	Information characteristics	granularity discrete/continuous modality (visual or auditory) complexity representation richness anticipated value synchronizaton context relevance
reduce stress emote humor cultivate enjoyment augment meaning or presence increase feeling of security	Satisfaction overall enhancement and approval of the general computing experience		

Abbildung 9: Aufmerksamkeits-Nutzen Tabelle nach McCrickard und Chewar²⁵

So ist es unter anderem denkbar, dass ein Mensch erhöhte Aufmerksamkeits-Kosten für eine Notification aufbringen muss, wenn dieser nicht hinreichende Erfahrungen²⁶ besitzt, um die Notification zu interpretieren oder die Notification an sich zu komplex formuliert wurde (siehe Information characteristics – complexity). Unter Berücksichtigung dieser Kostenfaktoren und dem daraus gewonnenen Nutzen lassen sich nach McCrickard und Chewar Anforderungen an ein Notification System definieren, wie der Abbildung 10 zu entnehmen.

²⁵ McCrickard D. 2003, S.68

²⁶ Siehe gemäß der Tabelle die User characteristic „Skills“

user notification goal	recommended display	not recommended
low interruption • minimal attention reallocation from primary task high reaction • make decisions, provide response low comprehension • long-term knowledge gain unimportant	in-place animations (blast and fade), small-sized	scrolling animations (ticker), large-sized
low interruption • minimal attention reallocation from primary task low reaction • no immediate response high comprehension • understand patterns and resources over time	scrolling animations (ticker), fast update	in-place animations (blast and fade), slow update

Abbildung 10: Mögliche Gliederung der user notification goals sowie angemessene Darstellungen für Notifications²⁷

Anhand dieser Matrix werden zwei mögliche Szenarien an Ausprägungen der zuvor vorgestellten user notification goals (interruption, reaction, comprehension) abgebildet. Jedes dieser Szenarien stellt dabei unterschiedliche Anforderungen an die Darstellung von angemessenen Notifications dar. So definiert das erste, in Abbildung 4 vorgestellte Szenario, beispielsweise die Benachrichtigungs-Ziele eines Nutzers, bei dem dieser eine möglichst geringe Unterbrechung seiner Aufgabe vornehmen muss, ein geringes Maß an Verständlichkeit benötigt wird, er aber dennoch eine Reaktion auf die Benachrichtigung vornehmen sollte. Für dieses Szenario bietet sich die Darstellung der Benachrichtigung in Form von kleinen Animationen auf der Benutzeroberfläche an, in der die Informationen vermittelt werden.

4.1.3 Awareness

Bei der Entwicklung von CSCW-Systemen, die eine kollaborative verteilte Aufgabenbewältigung realisieren sollen, ist eine intensive Auseinandersetzung mit dem Begriff der Awareness²⁸ meist unausweichlich. Nun kann man sich die Frage stellen, weshalb der Begriff in diesem Kontext derart relevant ist, dass sich mittlerweile eine Vielzahl an wissenschaftlichen Ausarbeitungen mit diesem befassen und versuchen diesem Begriff eine eindeutige Definition zuzuordnen. Tatsächlich ist es nicht trivial, eine einheitliche Definition dieses Begriffes zu finden, die in jedem Kontext gleichermaßen gültig ist. Um dem Begriff je nach Einsatzgebiet gerecht zu werden, wird

²⁷ McCrickard D. 2003, S.70

²⁸ dt.: Bewusstsein

dieser stets durch einen zusätzlichen Begriff präzisiert um eine angemessene Definition formulieren zu können. So trifft man bei einer Recherche des Awareness-Begriffs unter anderem auf die „Situational Awareness“ die „Passive Awareness“ oder „Workspace Awareness“. Nach einer ausführlichen Betrachtung verschiedener Definitionen mit einem Fokus auf diese Arbeit, bietet sich unter anderem eine allgemeine Definition des Awareness-Begriffs nach Dourish und Bly²⁹ an:

“Awareness involves knowing who is “around”, what activities are occurring, who is talking with whom; it provides a view of one another in the daily work environments. Awareness may lead to informal interactions, spontaneous connections, and the development of shared cultures – all important aspects of maintaining working relationships which are denied to groups distributed across multiple sites.”

Eine der größten Herausforderung bei der Entwicklung verteilter Kooperationssysteme besteht unter anderem darin, die wesentlichen Artefakte zu vermitteln, die Informationen über den aktuellen Gruppenkontext sowie den Aktivitäten der Gruppenteilnehmer enthalten. Arbeiten Menschen kollaborativ und physisch eng gekoppelt³⁰ an einer Aufgabe, so ergeben sich innerhalb dieser Gruppe Artefakte, die den einzelnen Teilnehmern Informationen über die Aktionen der anderen Gruppenteilnehmer vermitteln. Diese Artefakte werden dabei durch die Gruppenteilnehmer wahrgenommen und ermöglichen es ihnen, ein Bewusstsein über den aktuellen Kontext der Gruppe sowie der einzelnen Teilnehmer zu entwickeln. Dieses Bewusstsein bezeichnet einen wesentlichen Faktor in kollaborativen Situationen und fördert eine effizientere Aufgabenbewältigung innerhalb der Gruppe. Die Aufnahme solcher Artefakte sowie die Entwicklung eines Bewusstseins über die Gruppenaktivitäten, geschehen dabei im Optimalfall ohne dass der Mensch aktiv die Aufmerksamkeit von seiner primären Aufgabe ablenken muss. Bei einer genaueren Betrachtung der soeben aufgeführten Aspekte wird deutlich, dass dem Begriff Awareness als einzelne Entität noch keine hinreichende Bedeutung zugesprochen werden kann. Diese Eigenschaft identifizierte auch Kjeld Schmidt und machte dies in seiner Ausarbeitung deutlich:

²⁹ Bly H. 1993

³⁰ Z.B. in gemeinsam in einem Raum

“The first step towards some kind of conceptual clarity is to realize with the philosophers, from Husserl and Schutz to Wittgenstein and Ryle, that it does not make sense to conceive of ‘awareness’ as such, i.e., as a distinct (mental) entity. That is, the term ‘awareness’ is only meaningful if it refers to a person’s awareness of something.”³¹

Die Awareness bezieht sich nach Schmidt’s Zitat stets auf das Bewusstsein einer bestimmten Person gegenüber bestimmter Sachverhalte aus seiner Umwelt³². Möchte man für die Konzeption gebrauchstauglicher Benachrichtigungen für CSCW-Systeme Awareness-Aspekte berücksichtigen, so ist es ratsam, sich dieser Eigenschaft bewusst zu werden und eindeutig zu benennen, worauf sich die Awareness im jeweiligen Kontext beziehen sollte.

4.1.4 Awareness in einer verteilten Simulationsumgebung

Da sich der Begriff der Awareness stets auf eine bestimmte Domäne bezieht, ist es auch im Kontext einer verteilten Simulationsumgebung sinnvoll, die Relevanz des Begriffs und die zu berücksichtigenden Awareness-Aspekte zu identifizieren. Wie in Kapitel 4.1.1. erwähnt, besteht für solche Systeme ein besonderer Schwerpunkt darin, die Artefakte die innerhalb einer lokalen Gruppensituation zur Bewusstseins-Vermittlung vorliegen, auf ein verteiltes CSCW-System zu übertragen. Um die relevanten Awareness-Aspekte für diese Arbeit identifizieren zu können, wurden verschiedene Kreativitätstechniken, wie die KJ-Methode³³ angewandt. Die dabei erhobenen Awareness-Aspekte lassen sich wie folgt beschreiben:

Teilnehmerstatus

Die einzelne Person sollte stets ein Bewusstsein über die Verfügbarkeit der anderen Gruppenteilnehmer entwickeln können. Dadurch kann dieser unter anderem einschätzen, wann er andere Teilnehmern der Gruppe kontaktieren kann. Zu dem Teilnehmerstatus gehört z.B. der Online-Status eines Nutzers oder ob dieser aktuell verfügbar bzw. beschäftigt ist und nicht gestört werden möchte.

Teilnehmeraktivität

Die Vermittlung der Teilnehmeraktivitäten innerhalb einer Gruppe stellt Informationen

³¹ Schmidt K. 2002, S.287

³² Wie Sachverhalte, andere Personen, etc.

³³ Benannt nach dem Anthropologen Jiro, auch bekannt unter dem Begriff „Pinnwandmoderation“

über Handlungen einzelner Teilnehmer bereit. Dies ist bei einer kollaborativen Aufgabenbewältigung notwendig, da sich die Teilnehmer durch dieses Bewusstsein bei der Bearbeitung von Artefakten besser synchronisieren können. Zu Teilnehmeraktivitäten zählen neben der Bearbeitung von Artefakten durch einen Teilnehmer oder das zur Verfügung stellen von Artefakten auch Aktivitäten, in denen ein Teilnehmer bewusst Unterstützung bei der Bearbeitung eines Artefaktes benötigt.

Gruppenaktivität

Die Gruppenaktivität als Awareness-Aspekt definiert die allgemeinen Informationen bezüglich einer Gruppe. Die Gruppenaktivität kann dabei zum Teil aus den Teilnehmer-Zuständen und den Teilnehmeraktivitäten aggregiert werden. Neben Aktivitäten wie dem Ein- oder Austritt von Teilnehmern der Gruppe, kann hierbei auch der allgemeine Fortschritt bei der kollaborativen Aufgabenbewältigung adressiert werden.

Systemstatus

Neben Awareness-Aspekten, die den Fokus auf die Gruppendynamik legen, ist es auch sinnvoll den Teilnehmern ein Bewusstsein über den Systemstatus zu vermitteln. Dieser Aspekt ist besonders unter Berücksichtigung des Simulations-Kontextes sinnvoll, in dem das System nicht allein als Hilfsmittel eingesetzt wird um eine kollaborative Situation verteilt zu realisieren, sondern das System selbst Artefakte generieren kann. Die Verarbeitung und Generierung von Artefakten durch das System sollte daher in einer verteilten Simulationsumgebung in geeignetem Maße vermittelt werden.

4.2 Ereignisse in einer verteilten Simulationsumgebung

In verteilten Systemen, wie der verteilten Simulationsumgebung RCE, treten verschiedene Ereignisse auf, die es bei einer kollaborativen Aufgabenbewältigung zu berücksichtigen gilt. Neben lokal ausgelösten Ereignissen, die sich primär auf die Interaktion zwischen einem Nutzer und dem entsprechenden System beziehen, stellt insbesondere die Auseinandersetzung von Ereignissen in einem verteilten Kontext eine große Herausforderung bei der Erstellung eines geeigneten Konzeptes dar. Um einschätzen zu können, welche Ereignisse in welcher Form den Nutzern des verteilten Systems zugänglich gemacht werden, ist es zunächst erforderlich, diese angemessen zu klassifizieren. Anhand dieses Vorgehens lassen sich dann Benachrichtigungen gemäß dieser Klassifizierung von Ereignissen über geeignete Kommunikationskanäle/Darstellungsformen vermitteln. Allgemein stellt sich dabei allerdings zunächst die Frage, was ein Ereignis auszeichnet und gemäß

welchen Parametern sich dieses klassifizieren lässt. Durch die Identifikation aller benötigten Parameter kann gewährleistet werden, dass eine qualitativ hochwertige Erhebung aller Ereignis-relevanten Informationen erreicht wird.

4.2.1 Taxonomie nach McGrenere

Um eine Klassifikation vornehmen zu können, wurden verschiedene Konzepte und Ausarbeitungen betrachtet, die sich bereits ausführlicher mit dieser Thematik befassen. So sei an dieser Stelle unter anderem die Taxonomie von Ereignissen nach Joanna McGrenere³⁴ zu erwähnen, anhand derer sich eine erste Klassifizierung von Ereignissen gemäß bestimmter Kriterien vornehmen lässt. McGrenere differenziert dabei grundsätzlich zwischen zwei Faktoren:

Wer initiiert das Ereignis

Dieser Faktor gibt an, von wem ein Ereignis ausgelöst wird. McGrenere trifft hierbei eine Unterscheidung zwischen einem Individuum oder einer Gruppe.

Wie wird das Ereignis initiiert

Dieser Faktor gibt an, wie ein Ereignis von einem Initiator ausgelöst werden kann. Nach McGrenere erfolgt dies entweder direkt oder indirekt.

Wie wird ein Ereignis initiiert	Wer initiiert das Ereignis	
	Individuum	Gruppe
direkt	Nutzer löst Ereignis bewusst aus	n/a
indirekt	Nutzer löst Ereignis unbewusst aus	Gültig für eine CDE; Team-Prozess-Optimierungen

Tabelle 1: Taxonomie nach McGrenere³⁵

³⁴ McGrenere J. 2010, S.72

Anzumerken sei hierbei jedoch, dass die von McGrenere vorgestellte Taxonomie nicht 1:1 auf beliebige Domänen angewandt werden kann, da diese im Kontext eines bestimmten Projektes erstellt wurde. Dennoch eignen sich Teile dieser Methodologie auch im Kontext dieser Ausarbeitung für eine Identifikation sowie einer Gruppierung der auftretenden Ereignisse in RCE. So ist es auch für die Ereignis-Definition in RCE sinnvoll, eine Unterscheidung zwischen direkt und indirekt ausgelösten Ereignissen zu treffen. Betrachtet man die Anforderungen an das zu erstellende Konzept aus der Nutzungskontextanalyse, so wird deutlich, dass einige der dort spezifizierten Anforderungen ebenfalls direkt bzw. indirekt durch einen Nutzer vorgegeben werden.

Die in dem Anforderungskatalog spezifizierte Anforderung „1.1.1.4. Fehlermeldungen eines Tools senden“ beispielsweise, setzt eine bewusste Interaktion des Nutzer zum Initiieren eines Ereignisses voraus, indem der Nutzer aktiv z.B. eine entsprechende Benachrichtigung an den Tool-Entwickler übermittelt.

Das indirekte initiieren eines Ereignisses wird hingegen unter anderem durch die Anforderung „1.1.1.7. Tool-Verfügbarkeit anzeigen“ abgedeckt. In dieser wird aufgrund einer durch den Nutzer durchgeführten Aktion (z.B. veröffentlichen eines Tools) unbewusst der Status des Systems verändert, was zu einem Ereignis und einer Benachrichtigung anderer Nutzer in der verteilten Entwicklungsumgebung führen kann.

4.2.2 Anpassung der Taxonomie

Bei einer genaueren Betrachtung der Taxonomie wird deutlich, dass sich die Unterteilung der Initiatoren eines Ereignisses nicht gleichermaßen für alle verteilten Systeme anbietet. So können in einer verteilten Simulationsumgebung wie RCE auch Ereignisse auftreten, die sich nicht trivial auf eine (direkte oder indirekte) Nutzer-Interaktion zurückführen lassen. Diese Interaktionen werden in der Regel durch eine Änderung des System-Status initiiert, die durch das System selbst oder durch externe Ereignisse hervorgerufen wird. So ist es unter anderem möglich, dass in RCE aufgrund eines technischen Server-Defektes, verteilte Tools des Servers nicht mehr in der Simulationsumgebung vorhanden sind. Ein solches Ereignis wird weder direkt noch indirekt durch den Nutzer initiiert, sollte aber in jedem Falle berücksichtigt werden. Gemäß der bestehenden Taxonomie nach McGrenere lassen sich solche Ereignisse also nicht oder nur schwer abbilden. Aus diesem Grund wird eine Ergänzung der Taxonomie vorgeschlagen, in der das System als weitere relevante Komponente der Taxonomie enthalten ist:

Wie wird ein Ereignis initiiert	Wer initiiert das Ereignis		
	Individual	Gruppe	System
direkt	Nutzer löst Ereignis bewusst aus	n/a	System löst Ereignis direkt aus
indirekt	Nutzer löst Ereignis unbewusst aus	Gruppe löst Ereignis unbewusst über Monitoring des Systems aus	Ereignis durch externen Einfluss

Tabelle 2: Angepasste Taxonomie

Des Weiteren ist es je nach Kontext einer verteilten Simulationsumgebung möglich, dass nicht alle in der Taxonomie vorgestellten Initiatoren eines Ereignisses adressierbar sind. So ließen sich zum Zeitpunkt dieser Ausarbeitung in RCE keine Ereignisse identifizieren, die durch den Initiator „Gruppe“ ausgelöst werden. Da die vorgestellte Anpassung der Taxonomie nach McGrenere allerdings allgemein für CSCW-Systeme gültig sein soll und sich nicht nur auf RCE beschränkt, soll die Gruppe als Initiator auch weiterhin bestehen bleiben, um anderen Systemen gerecht zu werden.

4.2.3 Klassifikation der Ereignisse

Durch eine Anpassung der Taxonomie können nun die Ereignisse innerhalb einer verteilten Simulationsumgebung wie RCE benannt und entsprechend gruppiert werden. Dafür empfiehlt es sich, die Artefakte aus der Phase einer zuvor durchgeführten Nutzungskontextanalyse zu berücksichtigen und aus diesen gemäß der Taxonomie Ereignisse zu identifizieren. Die im Rahmen dieser Ausarbeitung resultierenden Ereignisse bestehen dabei jedoch nicht ausschließlich auf Basis eines Anforderungskataloges, sondern enthalten auch jene Ereignisse, die im Laufe der weiteren Entwurfsphase eines Konzeptes und Gestaltungslösungen nachträglich identifiziert wurden. Es ist daher grundlegend ratsam, eine Identifikation relevanter Ereignisse je nach System, nicht allein auf die Erkenntnisse einer abgeschlossenen

Nutzungskontextanalyse zu stützen, da durch die Erstellung erster Gestaltungslösungen Visionen entstehen können, die weitere relevante Ereignisse identifizieren. Dies wird unter anderem auch durch die Klassifizierung der Ereignisse nach der angepassten Taxonomie deutlich. So enthält diese Klassifizierung auch Ereignisse, die nicht unmittelbar aus Artefakten der Nutzungskontextanalyse resultieren. Die einzelnen Kategorien sind dabei wie folgt definiert:

Nutzer löst Ereignis bewusst aus

Der Nutzer des Systems steuert bewusst die Ereignisse des Systems und bestimmt somit selber welche Benachrichtigung an ein Individuum oder eine Gruppe gesendet wird. Hierzu zählt im Kontext von RCE unter anderem das der Nutzer bewusst eine auftretende Tool-Fehlermeldung an den Tool-Integrator sendet oder dieser anfragt, wann ein nicht verfügbares Tool wieder verfügbar ist.

Nutzer löst Ereignis unbewusst aus

Der Nutzer steht in einer Interaktion mit dem System und generiert durch seine Aktivitäten Ereignisse, die einem Individuum oder einer Gruppe vermittelt werden, ohne das der Nutzer diese bewusst kommunizieren muss. Im Kontext von RCE wäre dies unter anderem der Fall, wenn ein Tool-Integrator ein Tool veröffentlicht und die anderen Nutzer über die Veröffentlichung informiert werden oder wenn ein Nutzer einen Workflow startet, der dann durch andere Nutzer einsehbar ist.

System löst Ereignis selbstständig und direkt aus

Das System löst Ereignisse ohne zusätzliche Nutzer-Interaktion aus und vermittelt diese einem Individuum oder einer Gruppe. Insbesondere im Kontext einer verteilten Simulationsumgebung können dabei Artefakte auftreten, die durch das System generiert und den Nutzern kommuniziert werden müssen. Hierzu zählt unter anderem der automatische Verbindungsaufbau zu RCE-Instanzen auf Basis zuvor definierter Verbindungseinstellungen.

System löst Ereignis durch externen Einfluss aus

Je nach Kontext in dem sich das betrachtete verteilte System befindet ist es möglich, dass der Zustand des Systems durch externe Einflüsse manipuliert wird. Dies kann neben technischen Problemen wie beispielsweise mangelnder Speicherplatz für weitere generierte Daten oder Verbindungsproblemen auch die Änderung des Systemzustandes über externe Schnittstellen beinhalten. Da solche externen Einflüsse dem Nutzer

mitunter nur schwer zugänglich sind, sollten diese hinreichend kommuniziert werden, damit der Nutzer nachvollziehen kann, wodurch ein solches Ereignis ausgelöst wurde.

4.3 Benachrichtigungen in einer verteilten Simulationsumgebung

Dieses Kapitel befasst sich mit dem Begriff der Benachrichtigungen im Kontext einer verteilten Simulationsumgebung

4.3.1 Begriffsdefinition – Benachrichtigung

Da der Begriff „Benachrichtigung“ allgemein viel Freiraum für Interpretationen zulässt, ist zunächst eine entsprechende Definition im Kontext dieser Ausarbeitung sinnvoll. Anhand der hier vorgestellten Definition soll so ein grundlegendes Verständnis über den Begriff vermittelt und abweichende Interpretationen im Kontext dieser Ausarbeitung vermieden werden. Das Finden einer angemessenen Definition des Benachrichtigungsbegriffs auf Basis vorangegangener Arbeiten ist dabei nicht trivial, da oftmals ein Verständnis dieses Begriffs vorausgesetzt und daher nicht ausführlich definiert wird. Dennoch kann der Begriff je nach Kontext zu unterschiedlichen Interpretationen führen, weshalb dieser stets eindeutig zu formulieren ist. Während in verschiedenen Arbeiten eine Benachrichtigung stets als textuelle Informationen betrachtet wird, sind andere Definitionen relativ offen formuliert und beschränken den Begriff nicht auf konkrete Darstellungsformen, sondern definieren Benachrichtigung allein über deren Nutzen. Letzteres kann unter anderem durch die Definition nach Haifeng Shen et al.³⁶ verdeutlicht werden, in der auch die Relevanz des Begriffs für kollaborative Systeme nahe gelegt wird:

“Notification is an essential feature in collaborative systems, which determines when, what, and how updates made by one user are propagated, applied, and reflected on other users’ interfaces. It is this feature that distinguishes collaborative systems from traditional multi-user systems, such as database management systems and timesharing operating systems, where a user is normally not notified of the actions performed by other users.”

³⁶ Shen H. 2002, S.1

Grundsätzlich bezeichnet der Begriff der Benachrichtigungen demnach einen Träger von Informationen über verschiedenste Ereignisse, die dem Nutzer eines Systems vermittelt werden. Ob die Benachrichtigung Informationen nun rein textuell oder unter Verwendung anderer Darstellungsformen vermitteln, sollte dabei stets unter Berücksichtigung des jeweiligen Kontextes definiert werden. Möchte man innerhalb der Konzeption einer gebrauchstauglichen Ereignis-Behandlung in einem CSCW-System Rücksicht auf Awareness-Aspekte nehmen, wie dies in Kapitel 4.1.3. angesprochen wurde, so sollten in jedem Fall auch Informationen die rein auf grafischer Ebene kommuniziert werden als Benachrichtigung betrachtet werden. Gemäß der Betrachtung verschiedener Definitionen und unter Berücksichtigung soll im Rahmen dieser Ausarbeitung folgende Definition des Benachrichtigungs-Begriffes vorgenommen werden:

Eine Benachrichtigung im Kontext von CSCW- bzw. CSCL-Systemen bezeichnet den Träger von Informationen, um die Nutzer des jeweiligen Systems über auftretende Ereignisse zu informieren. Diese Ereignisse können sich dabei sowohl auf Änderungen von System-Komponenten als auch auf Änderungen der Gruppendynamik beziehen. Dabei sind Benachrichtigungen an keine bestimmte Darstellungsform oder an einen Kommunikations-Kanal gebunden, sondern können neben textuellen Darstellungen auch rein durch grafische oder audiovisuelle Methoden vermittelt werden.

4.3.2 Darstellungsformen für Benachrichtigungen

Dieses Kapitel befasst sich mit einer Betrachtung gängiger Darstellungsformen für Benachrichtigungen, die sich im Zuge der Entwicklung von Software-Systemen und der allgemeinen Mensch-Computer-Interaktion entwickelt und zunehmend etabliert haben. Darüber hinaus wird auch auf einige Konzepte neuartiger Darstellungsformen eingegangen. Die Abwägung, welche Darstellungsformen an Benachrichtigungen letztendlich in diesem Kapitel behandelt werden und welche nicht, basiert dabei auf eigenen Erfahrungen und der Berücksichtigung verschiedener Style-Guides³⁷ und einer Betrachtung diverser Software-Systeme³⁸. Es ist daher nicht auszuschließen, dass durchaus noch weitere Darstellungsformen existieren, die in dieser Ausarbeitung nicht berücksichtigt wurden.

³⁷ Z.B. Style-Guide für Android-Notifications, Windows “Toast”-Notifications etc.

³⁸ Wie Microsoft Windows, Apple’s iOS, Android, Linux

Balloon-Messages

Balloon Messages³⁹ werden klassisch über den Desktop eines Betriebssystems angezeigt und sind nur für einen kurzen Zeitraum sichtbar (siehe Abb. 11). Die Darstellung einer Balloon-Message ist dabei in der Regel abhängig vom jeweils eingesetzten Betriebssystem, wird jedoch stets am Bildschirmrand dargestellt, um den Nutzer bei einer Interaktion mit dem System oder seiner Aufgabenbewältigung nicht zu unterbrechen. Dementsprechend erfordert der Einsatz von Balloon-Messages zur Darstellung von Ereignis-Informationen auch keine direkte Interaktion mit diesen. Optional kann der Nutzer jedoch durch einen Klick auf die Balloon-Message in eine Interaktion treten, um sich so weitere Informationen anzeigen zu lassen oder um zu dem entsprechenden Ereignis zu navigieren. Die Interaktions-Möglichkeiten einer Balloon-Message können dabei je nach Betriebssystem teils stark variieren. Die Darstellung einer Balloon-Message erfolgt zudem stets im Vordergrund und überdeckt die auf dem System geöffneten Anwendungen. Unter anderem aus diesem Grund, fällt der reine Informationsgehalt einer Balloon-Message vergleichsweise gering aus, um möglichst wenig Platz in Anspruch zu nehmen und keine, für den Nutzer relevante, Inhalte zu überdecken. Auffallend ist zudem, dass sowohl das Erscheinen als auch das Verschwinden einer Balloon-Message unter gängigen Betriebssystemen nicht abrupt, sondern stets unter Verwendung einer Blenden-Animation erfolgt, sodass die Aufdringlichkeit dieser möglichst gering gehalten wird.

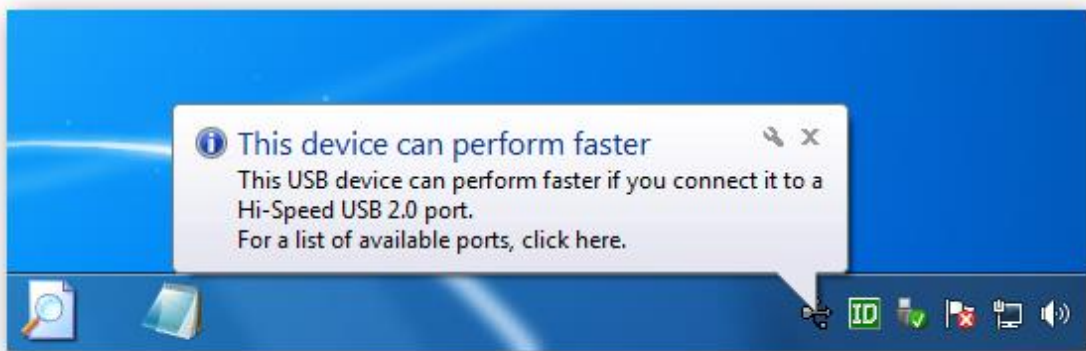


Abbildung 11: Balloon-Message unter Windows 7

Popup-Fenster

Ähnlich der Balloon-Messages, werden auch Popup-Fenster gemäß des jeweiligen Betriebssystems dargestellt. Jedoch erfolgt die Darstellung von Ereignis-Informationen hierbei über ein eigens erzeugtes Fenster, weshalb eine Interaktion durch den Nutzer meist vorausgesetzt wird. Je nach Betriebssystem oder Anwendung, kann der Nutzer

³⁹ dt.: Balloon-Benachrichtigungen

dabei keine weitere Interaktion mit dem System oder der entsprechenden Anwendung vornehmen, solange er nicht das Popup-Fenster geschlossen hat. Die Interaktionsmöglichkeiten variieren hierbei aufgrund der flexibel wählbaren Größe eines Popup-Fensters und können neben simplen Bestätigungs-Buttons auch komplexere Dialoge beinhalten. Neben rein textuellen Informationen, werden Popup-Fenster oftmals mit grafischen Symbolen angereichert, um gegebene Informationen den Nutzern optisch kategorisiert darstellen zu können. Die Darstellung von Popup-Fenstern erfolgt, im Gegensatz zu Balloon Messages und weiteren Darstellungsformen, meist zentriert auf der Oberfläche des jeweiligen Betriebssystems. Der Einsatz von Popup-Messages für die Darstellung von Ereignis-Informationen sollte daher ausgiebig durchdacht werden, da diese sehr aufdringlich sind und die Aufmerksamkeit eines Nutzers vollständig beanspruchen. Daher ist eine hinreichende Abwägung gemäß des „Aufmerksamkeits-Kosten-Modells nach McCrickard et.al. zu empfehlen (siehe Kapitel 4.1.2.2.).

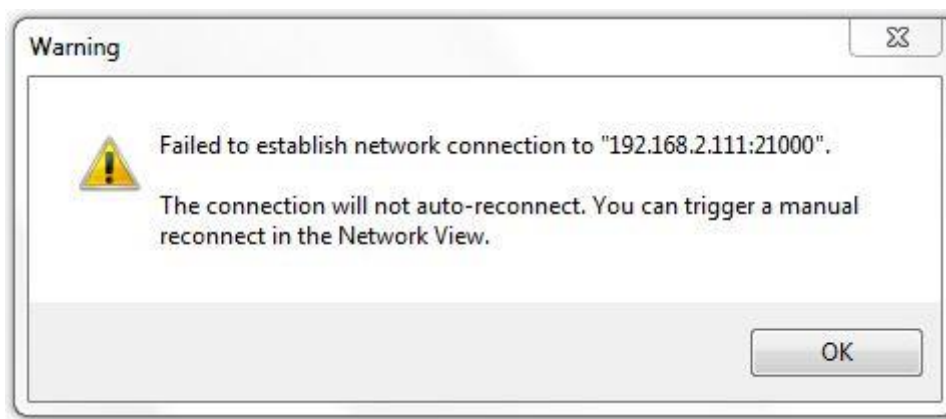


Abbildung 12: Warnmeldung in Form eines Popups in RCE

Mouseover-Information (Tool-Tip)

Die Mouseover-Information ermöglicht es dem Nutzer, Informationen über bestimmte Objekte oder Funktionalitäten zu erhalten, indem er mit dem Mauszeiger eine gewisse Zeit⁴⁰ über dem jeweiligen Objekt verweilt. Mouseover-Informationsfelder stellen in der Regel einen nur stark eingeschränkten Platz für Informationen zu Verfügung und können daher nur kurze textuelle Informationen enthalten. Im Kontext von Ereignis-Informationen lassen sich Mouseover-Informationen beispielsweise als zusätzliche Informationsquelle einsetzen, um Status-Änderungen von Objekten zu vermitteln. So lassen sich Benachrichtigungen grafischer Form, durch die Änderung eines Icons,

⁴⁰ In der Regel ca. 1-2 Sekunden

zusätzlich mit Informationen anreichern. Eine Interaktion mit Mouseover-Informationen ist nicht vorgesehen bzw. auch nur bedingt realisierbar, da die Information nur so lange aktiv dargestellt wird, solange der Cursor der Maus nicht von dem jeweiligen Objekt bewegt wird. Für die Darstellung von Mouseover-Informationen ist eine direkte Interaktion durch den Nutzer notwendig, da dieser bewusst die Objekte mit dem Cursor ansteuern muss, um sich die Informationen anzeigen zu lassen. Dieses Vorgehen mag dem allgemeinen „Charakter“ einer Benachrichtigung widersprechen, da Benachrichtigungen in der Regel ohne eine bewusste Interaktion durch den Nutzer auf verschiedene Art dargestellt werden. Allerdings kann durch den bewussten Einsatz von Mouseover-Informationen die Darstellung von Ereignis-Informationen, insbesondere grafischer Art, unterstützt werden, weshalb diese im Kontext dieser Ausarbeitung als eigene Darstellungsform berücksichtigt werden.

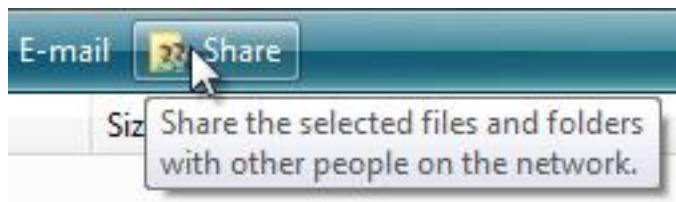


Abbildung 13: Darstellung eines Tool-Tips unter Windows

Notification-Collector

Mittlerweile existiert eine Vielzahl an Systemen, in denen es erforderlich ist, eingehende Notifications zu sammeln und den Nutzern auch persistent zur Verfügung zu stellen. Der Begriff des „Notification-Collector“ wird daher im Rahmen dieser Ausarbeitung für eben erwähnte Funktionalität gewählt, da die Bezeichnung einer solchen Komponente von System zu System variieren kann. Eine der wohl bekanntesten Begrifflichkeiten ist unter anderem die des „Notification-Centers“ und findet insbesondere in mobilen Endgeräten wie Android oder iOS-Devices Einsatz. Die Darstellung der Notifications kann dabei in verschiedene Kategorien unterteilt und durch den Nutzer teilweise individualisiert werden. Der Informationsgehalt eines solchen Notification-Collectors ist im Vergleich zu einzelnen Notification-Darstellungen relativ hoch, kann allerdings je nach Frequenz der eingehenden Informationen schnell an Übersichtlichkeit verlieren.

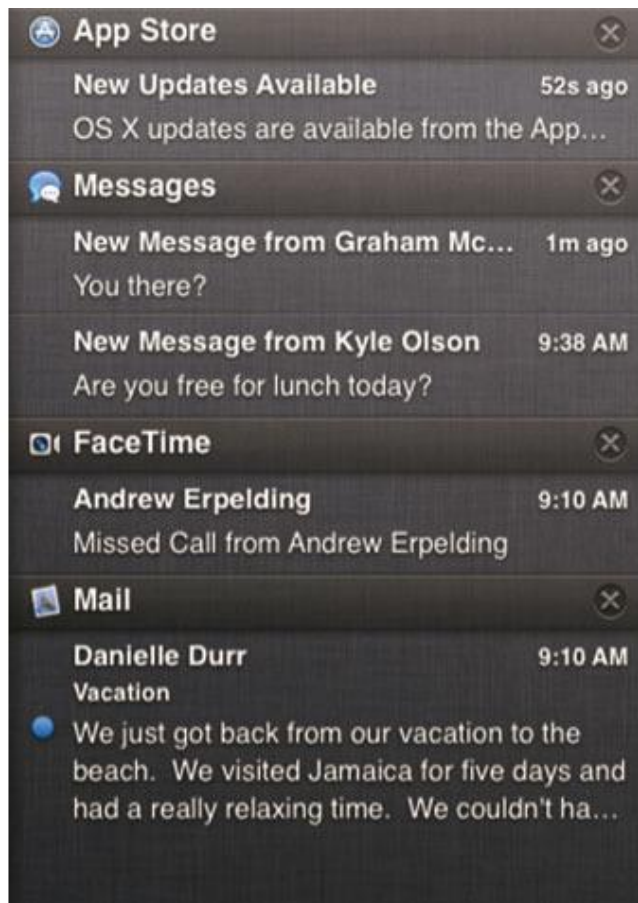


Abbildung 14: Darstellung eines Notification-Centers unter iOS 6⁴¹

Log-Darstellungen

In verschiedenen Systemen, wie beispielsweise Entwicklungsumgebungen, ist eine Darstellung von relevanten Log-Benachrichtigungen essentiell. So stellt auch RCE als Beispiel einer verteilten Simulationsumgebung verschiedene „Views“ bereit, in denen unter anderem Log-Einträge wichtiger Ereignisse hinterlegt sind. Neben den durch einen Benutzer bewusst entwickelten Log-Outputs verschiedener Tools oder Workflows, enthalten diese unter anderem auch Informationen über Fehlermeldungen von Workflows oder Tools. Darüber hinaus lässt sich ihnen entnehmen, ob ein Workflow erfolgreich abgeschlossen oder mit einem Fehler beendet wurde.

Notification-Bar

Bei der Notification-Bar handelt es sich um eine statisch platzierte Informationsquelle auf der Benutzeroberfläche. An dieser Stelle sei zu erwähnen, dass der Begriff „Notification-Bar“ Diese zeigt eingehende Benachrichtigungen anhand eines kurzen

⁴¹ Quelle: www.macworld.com

und möglichst aussagekräftigen Textes an. Gelangt eine neue Benachrichtigung an den Client des Nutzers, so wird die alte Benachrichtigung überschrieben und ersetzt. Die Notification-Bar wird dabei in der Regel am oberen Bereich einer Benutzeroberfläche angeordnet. Über die Notification-Bar kann somit ein gewisses Maß an Awareness realisiert werden.



Abbildung 15: Notification-Bar unter iOS⁴²

Grafische Status-Repräsentation

Der Einsatz von grafischen Status-Repräsentationen verzichtet bewusst auf textuelle Informationen und ermöglicht unter anderem die Darstellung der Änderung von Objektzuständen (Ereignis) durch eine Anpassung der zugehörigen Objekt-Grafiken. Neben diesen Zuständen lassen sich im Kontext eines CSCW-Systems zudem auch Informationen über die Nutzer-Aktivitäten eines verteilten Systems vermitteln. Diese Methode ist insbesondere dann von Vorteil, wenn es für eine kollaborative Aufgabenbewältigung erforderlich ist, dass die Nutzer ein Bewusstsein über die allgemeine Gruppendynamik entwickeln müssen. So wird unter anderem gewährleistet, dass sich die Nutzer eines verteilten Systems einander besser koordinieren können, um so eine effizientere Aufgabenbewältigung zu ermöglichen. Optional lassen sich solche Repräsentationen durch den Einsatz von MouseOver-Informationen zusätzlich anreichern. Mittlerweile lässt sich der Einsatz von grafischen Status-Repräsentationen

⁴² Quelle: www.stackoverflow.com

Der Einsatz von grafischen Status-Repräsentationen findet mittlerweile in einer Vielzahl an Systemen statt.

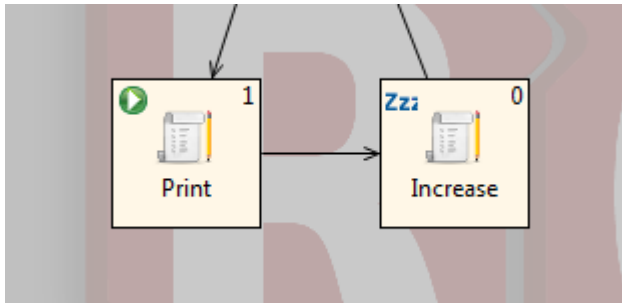


Abbildung 16: Darstellung von Tool-Zuständen in RCE. Links, Tool in Durchführung. Rechts, Tool "wartet" auf Input



Abbildung 17: Mail-Kopf mit Statusdarstellung in MS Outlook

Notification Interfaces nach Booker et. al.

In ihrer Ausarbeitung über Notification Interfaces stellen John E. Booker et. al. drei Darstellungsformen vor, um verschiedene Ereignis-Informationen als Metriken in einem grafischen Interface zu repräsentieren. Der Fokus dieser Arbeit liegt dabei insbesondere auf der Vermittlung von Gruppen-Aktivitäten innerhalb eines CSCW-Systems. Booker et. al. identifizieren für diese Interfaces insgesamt sechs Parameter, die in jeder dieser Darstellungen abgebildet werden sollen:

Population

Die Population gibt die Anzahl der Teilnehmer in einem CSCW-System dar.

Movement

Das Movement⁴³ definiert die Aktivität der Nutzer in drei Abstufungen. Darunter zählen

⁴³ dt. Bewegung

neben direkten Interaktionen mit einem CSCW-System auch physische Aktivitäten einzelner Nutzer, wie beispielsweise die Bewegung in einem Raum.

Location

Der Location-Parameter (dt. Position) beschreibt die aktuelle Position der einzelnen Gruppen-Teilnehmer eines CSCW-Systems.

Familiarity

Die Familiarity⁴⁴ ist ein Maß, welches die Vertrautheit der einzelnen Gruppenteilnehmer beschreibt. Treten einer Gruppe bislang unbekannte oder „neue“ Teilnehmer bei, so wird die Familiarity allgemein geringer eingestuft.

Collaborative Work

Dieser Parameter definiert, in wie fern Nutzer eines CSCW-Systems in einem aktuellen Kontext gemeinsam Aufgaben bewältigen oder ob diese eigenständig arbeiten

Ein Wesentlicher Schwerpunkt bei der Darstellung dieser gruppendynamischen Parameter bestand nun primär darin, diese den einzelnen Nutzern des CSCW-Systems gebrauchstauglich darzustellen, ohne dabei jedoch viel Platz auf der Benutzeroberfläche zu beanspruchen. Booker et. al. entwickelten daher unter anderem ein animiertes Gesicht (genannt „Smiling George“), welches die einzelnen Parameter anhand verschiedener Mimiken abbilden kann. So wurde unter anderem die Population einer Gruppe durch das Lächeln des Gesichts dargestellt oder das Movement durch eine Bewegung des Gesichts innerhalb des Fensters realisiert. Eine ausführliche Beschreibung dieser, und der weiteren zwei untersuchten Darstellungsformen soll an dieser Stelle allerdings nicht erfolgen. Interessierte Leser können für eine weitere Auseinandersetzung mit der Thematik jedoch die entsprechende Ausarbeitung von Booker et.al. zu Rate ziehen.

⁴⁴ Vertraulichkeit

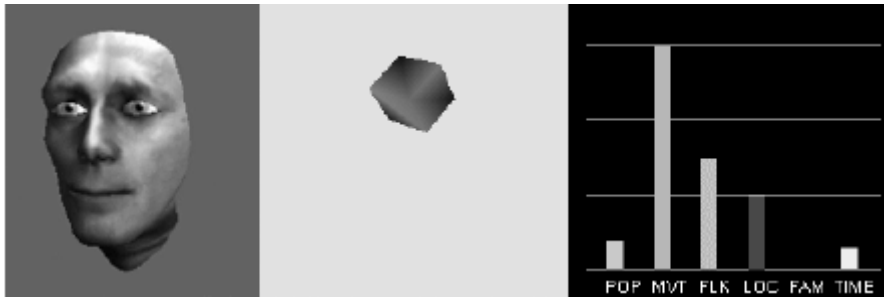


Abbildung 18: Notification-Interfaces nach Booker et.al Von links nach rechts: Smiling George, Spinning Cube und Activity Graph

Eine Betrachtung dieses Konzeptes wurde im Rahmen dieser Ausarbeitung als sinnvoll erachtet, da Booker et.al. insbesondere auf den Aspekt der Awareness eingehen und zudem identifizieren, welche Metriken für die Vermittlung von Gruppenkontext-relevanten Informationen sinnvoll erscheinen.

Scope Notification System nach Dantzich et. al.

Das „Scope Notification System“ von Maarten van Dantzich et.al.⁴⁵ beschreibt ein Interface zur Darstellung von Benachrichtigungen aus verschiedenen Informationsquellen in einer Gesamt-Übersicht. Das Design orientiert sich dabei an der typischen Form und Anordnung eines Radars und soll dem Nutzer des Notification-Systems so ermöglichen, kritische Benachrichtigungen anhand dieser metaphorischen Darstellung schnell zu identifizieren. Die Benachrichtigungen werden daher nicht zufällig auf dem Interface angeordnet, sondern liegen, je relevanter diese sind, weiter im Zentrum des Radars. Ziel dieses Konzeptes ist unter anderem ein Notification-System, das Awareness-Aspekte in einem kollaborativ verteilten System berücksichtigt. So soll der Nutzer durch den Einsatz des Notification-Systems ein Bewusstsein über eingehende Benachrichtigungen und deren Dringlichkeit entwickeln können, ohne dabei seine Aufmerksamkeit für einen längeren Zeitraum von seiner primären Aufgabe abzulenken. Durch dieses Konzept wird demnach auch berücksichtigt, dass die Aufmerksamkeit eines Nutzers eine eingeschränkte Ressource darstellt und demnach nicht für einen längeren Zeitraum auf das Notification-System gelenkt werden sollte. Dantzich et. al. definieren in ihrem Konzept insgesamt vier Kategorien, die sich im Kontext eines Informations- und Management-Systems ergeben. Je nach Einsatzgebiet ist es daher möglich, dass die Kategorien des Scope Notification Systems beliebig angepasst oder das System um zusätzliche Kategorien gemäß der Anforderungen des jeweiligen Kontextes angereichert wird. Auch sehen Dantzich et. al. die Möglichkeit vor, die einzelnen Kategorien in deren Relevanz anpassen zu können, indem wichtige

⁴⁵ Dantzich M. 2002, S.268

Kategorien mehr und weniger wichtige Kategorien weniger Platz auf dem Radar zugeteilt bekommen. Da innerhalb der identifizierten Kategorien jeweils selbst nochmal unterschiedliche Informationen dargestellt werden können⁴⁶ sieht das Konzept eine Identifikation dieser durch angemessene Icons auf dem Radar vor. So ist der Abbildung 19. zu entnehmen, dass in der Kategorie der „Inbox“ die Benachrichtigungen durch verschiedene Icons abgebildet werden. Jedes dieser Icons definiert dabei eine bestimmte Kategorie an eingehenden E-Mail-Notifications. So lassen sich beispielsweise E-Mails von Favoriten in Form eines Stern-Icons darstellen.

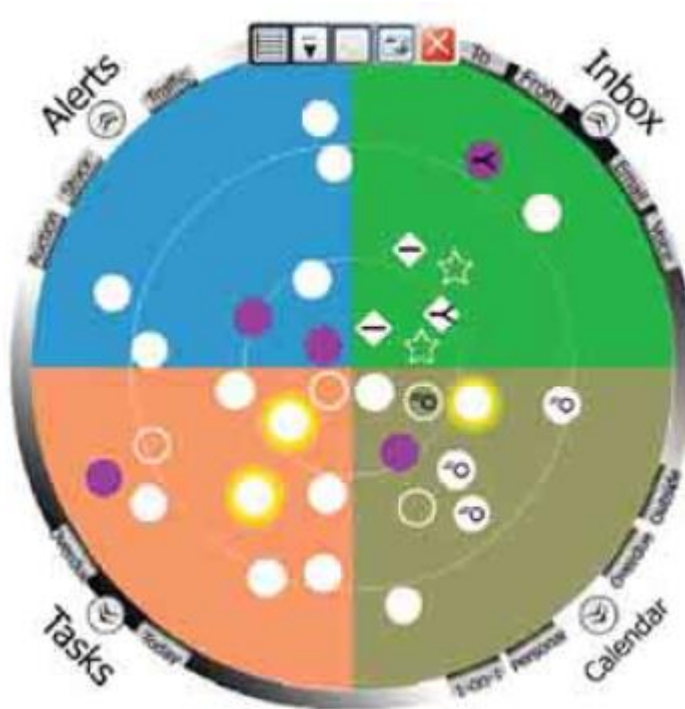


Abbildung 19: Scope Notification System nach Dantzig et.al. Besonders relevante Benachrichtigungen werden im Zentrum angezeigt

Ähnlich dem Ansatz nach Booker et.al. soll auch nach diesem Konzept ein Bewusstsein über eingehende Benachrichtigungen vermittelt werden. Das besondere hierbei ist jedoch, dass neben dem allgemeinen Eingang einer Information auch deren Dringlichkeit über eine metaphorische Darstellung stattfindet. Im Kontext der verteilten Simulationsumgebung RCE, ist daher die Überlegung sinnvoll, die Relevanz einer eingehenden Ereignis-Benachrichtigung ebenfalls durch eine angemessene Darstellungsform zu vermitteln. Durch ein solches Vorgehen könnte der Nutzer so

⁴⁶ Z.B. geschäftliche E-Mails, private E-Mails, Favoriten, etc.

effizient identifizieren, welche der eingehenden Ereignis-Benachrichtigungen besonders wichtig sind und welche er ggf. zu einem späteren Zeitpunkt lesen kann.

4.3.3 Klassifikation der Benachrichtigungen

Nach einer Betrachtung gängiger Darstellungsformen und der weiteren Konzepte für Benachrichtigungen wird deutlich, dass diese je nach Kontext des jeweiligen Systems unterschiedlichste Vor- sowie Nachteile beinhalten. Um nun ein geeignetes Konzept für die Behandlung und Darstellung von Ereignis-Informationen in einer verteilten Simulationsumgebung wie RCE entwickeln zu können, ist es zunächst erforderlich, diese Darstellungsformen einander abzuwägen und kritisch zu diskutieren. Nun stellt sich allerdings die Frage, nach welchen Kriterien können Benachrichtigungen überhaupt klassifiziert werden? Welche Faktoren spielen bei einer Abwägung eine wesentliche Rolle? Betrachtet man die letzten Abschnitte dieses Kapitels etwas genauer, so wird deutlich, dass sich die wesentlichen Faktoren zum Teil aus den verschiedenen Darstellungsformen für Benachrichtigungen, sowie aus den relevanten Konzepten erheben lassen:

Informationsgehalt

Die Menge der zeitgleich darstellbaren Informationen, ob textuell oder grafisch, wird als Faktor durch den Informationsgehalt definiert. Je nachdem, wie umfangreich die Ereignis-Informationen ausfallen, die es zu vermitteln gilt, ist eine entsprechende Darstellungsform zu wählen, die den notwendigen Raum bietet, um die Informationen möglichst vollständig abbilden zu können.

Persistenz

Die Persistenz definiert, wie lange die Informationen über eine Benachrichtigung auf der Benutzeroberfläche dargestellt werden und ob diese, unter Umständen durch zusätzliche Interaktionen des Nutzers, auch dauerhaft gespeichert werden kann. So ist es unter anderem denkbar, dass ein Nutzer die dargestellten Informationen dauerhaft sichern oder exportieren möchte, sofern die Benachrichtigungs-Darstellung die einzelnen Benachrichtigungen nativ nur kurzzeitig darstellt.

Interaktionsmöglichkeit

Die Interaktionsmöglichkeit als Faktor gibt an, in wie fern der Nutzer in eine Interaktion mit der Benachrichtigung treten kann, um sich beispielsweise zusätzliche Informationen

anzeigen zu lassen oder Benachrichtigungen allgemein zu verwalten (z.B. das Schließen oder Speichern von Benachrichtigungen). Dieser Faktor hängt dabei oftmals unmittelbar mit der Aufdringlichkeit einer Benachrichtigung zusammen, wenn der Nutzer beispielsweise den Erhalt einer Benachrichtigung bewusst bestätigen muss-

Aufdringlichkeit

Ein Faktor der sich verhältnismäßig nur recht schwer erfassen lässt beschreibt die Aufdringlichkeit, die durch das Erscheinen einer Benachrichtigung ausgelöst wird. Dieser Faktor beschreibt, in welchem Maße die Aufmerksamkeit eines Benutzers auf die erscheinende Benachrichtigung gelegt wird.

Die identifizierten Faktoren können anschließend in einer übersichtlichen Matrix auf die zuvor vorgestellten Darstellungsformen für Benachrichtigungen abgebildet werden, um die Ausprägungen der Darstellungen einander effizienter abwägen zu können. Die komplette Matrix kann dem Anhang dieser Ausarbeitung entnommen werden.

4.3.4 Das Modell der Ereignisse und Benachrichtigungen

Nachdem in den vorangegangenen Kapiteln eine Betrachtung sowie angemessene Klassifizierung des Ereignis-Begriffs erfolgte und unterschiedliche Darstellungsformen an Benachrichtigungen ausführlicher diskutiert wurden, ist es für die Konzeption eines Benachrichtigungs-Systems sinnvoll, die aus diesen Kapiteln gewonnenen Erkenntnisse zu aggregieren, um daraus die Anforderungen an gebrauchstaugliche Benachrichtigungen erheben zu können. Um die Zusammenhänge aus Ereignissen und Benachrichtigungen besser nachvollziehen zu können, wurde auf Basis des RCE-Systems ein Modell erstellt, in dem die wesentlichen Komponenten eines Benachrichtigungs-Systems im Kontext einer verteilten Simulationsumgebung enthalten sind. Im Anschluss fand eine Generalisierung dieses Modells statt, um es allgemein auf verteilte Systeme abbilden zu können. Dieses Modell umfasst insgesamt vier essentielle Komponenten:

Nutzer

Die Nutzer treten in eine Interaktion mit dem verteilten System. Dabei steht neben der eigentlichen Interaktion mit dem System selbst, insbesondere die Kommunikation mit anderen Nutzern während der Aufgabenbewältigung im Fokus. So setzen die Nutzer das System ein, um eine physisch entkoppelte Gruppensituation zu realisieren, in der kollaborativ gearbeitet wird.

System

Das System stellt allgemein Funktionalitäten zur Verfügung, mit denen sich Artefakte für eine Aufgabenbewältigung erstellen und bearbeiten lassen. Diese Artefakte können je nach Domäne des eingesetzten Systems variieren. Im Kontext von CSCW-Systemen ist das System darüber hinaus verantwortlich, die Kommunikation zwischen verteilt agierenden Nutzern zu gewährleisten und relevante Informationen über den aktuellen Gruppenkontext den Nutzern des Systems angemessen zu vermitteln.

Ereignis

Die Ereignisse des Modells beziehen sich auf alle Ereignisse, die in Relation zu dem System stehen. Neben Ereignissen die durch die primären Nutzer des Systems initiiert werden, kann es auch externe Ereignisse geben, dessen Auslöser dem Nutzer oder dem System nicht bekannt sind. Darüber hinaus kann ein eingetretenes Ereignis selbst als Initiator weiterer Ereignisse fungieren. Das Auftreten von Ereignissen muss je nach Initiator und Nutzungskontext den Nutzern durch gebrauchstaugliche Notifications vermittelt werden.

Notification

Die Notifications informieren über auftretende Ereignisse innerhalb des CSCW-Systems und werden vom System durch die Ereignisse generiert. Die Informationen müssen dabei entsprechend aufbereitet werden, um diese dem Nutzer verständlich kommunizieren zu können.

Die identifizierten Relationen der einzelnen Komponenten in einem verteilten System können dem Modell in Abbildung 20 entnommen werden. Anhand der eingehenden Verbindungen der Ereignis-Komponente kann ein Ereignis nach diesem Modell durch insgesamt drei Initiatoren ausgelöst werden:

- Nutzer
- System
- Ereignis

Unter Berücksichtigung dieses Modells soll nun gewährleistet werden, die Initiatoren von Ereignissen innerhalb eines verteilten Systems eindeutig identifizieren zu können, um daraus anschließend Anforderungen an angemessene und gebrauchstaugliche Notifications zu erheben. Um dies zu ermöglichen, wurde die angepasste Taxonomie von McGrenere auf das Modell überführt und findet sich in insgesamt drei Kategorien

in dem Modell wieder. So beschreibt das Modell die Zusammenhänge von Ereignissen die

- direkt durch Nutzer,
- indirekt durch Nutzer,
- durch das System

initiiert werden. Auffallend bei diesem Modell ist, dass die „Gruppe“ als Initiator nicht unmittelbar enthalten ist. Daher sei an dieser Stelle anzumerken, dass die Komponente der „Nutzer“ sich nicht ausschließlich auf eine Instanz beziehen muss, sondern auch mehrere Nutzer respektive eine Gruppe beinhalten kann. Des Weiteren ist zu berücksichtigen, dass das vorgeschlagene Modell den Fokus auf die kausalen Zusammenhänge von Ereignissen in CSCW-Systemen legt und nicht auf eine Beschreibung der technischen Zusammenhänge abzielt. Dies wird unter anderem durch den Anwendungsfall „Nutzer initiiert Ereignis“ verdeutlicht, der dem Modell zu entnehmen ist. Aus einer rein technischen Betrachtung wäre dieser Anwendungsfall nicht korrekt, da der Nutzer - obwohl er ein Ereignis direkt initiiert - dies letztendlich durch eine Interaktion mit dem System realisiert. Gemäß einer technischen Betrachtung, müsste dieser Anwendungsfall also ebenfalls das System als Komponente beinhalten. Diese technische Betrachtung wurde in dem Modell jedoch bewusst nicht berücksichtigt, um den Unterschied zwischen bewusstem (direkt) sowie unbewusstem (indirekt) Initiieren von Ereignissen durch einen Nutzer zu verdeutlichen.

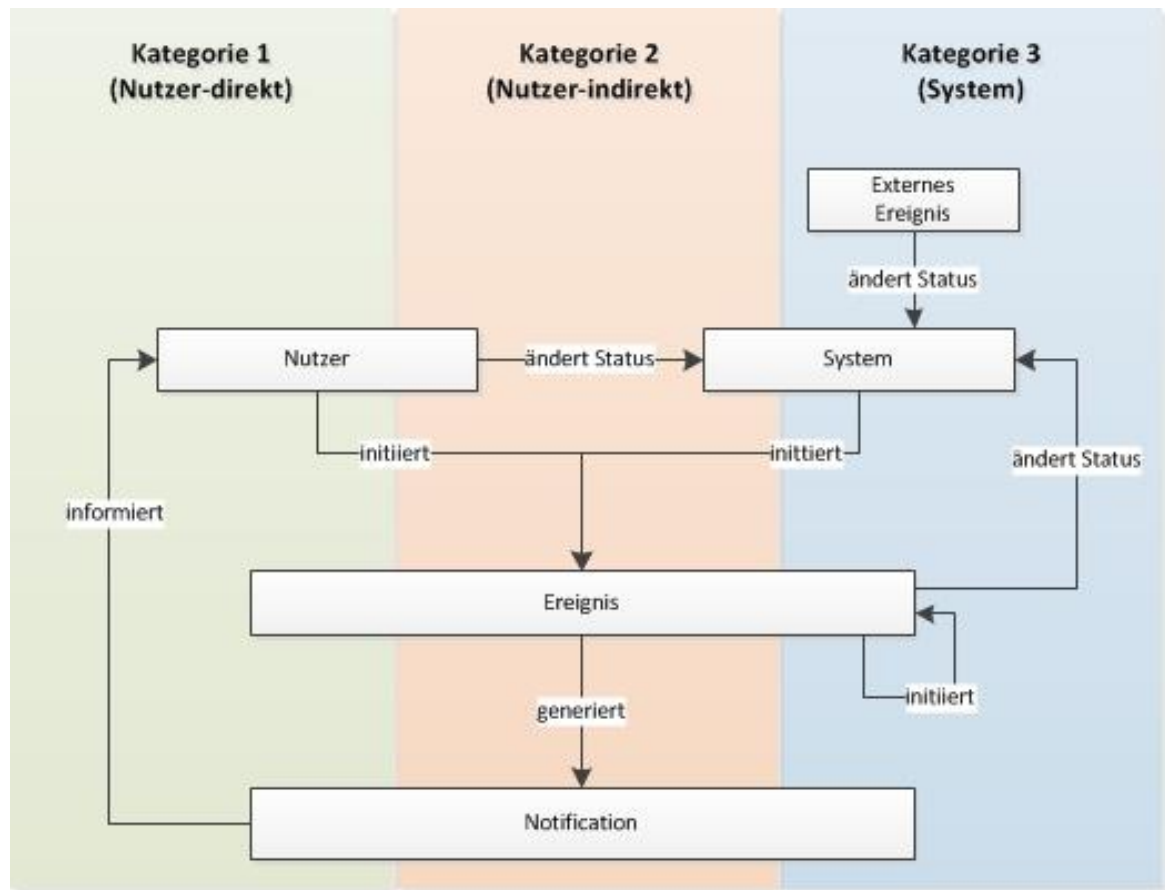


Abbildung 20. Modell mit Klassifizierung der Ereignis-Initiatoren

Nachdem die Ereignisse eines CSCW-Systems gemäß dem vorgestellten Modell klassifiziert wurden, lassen sich anschließend die Anforderungen an angemessene und gebrauchstaugliche Benachrichtigungen für diese Ereignisse erheben. Jede der in dem Modell beschriebenen Kategorien beinhaltet dabei spezifische Anforderungen, die bei dem Entwurf erster Gestaltungslösungen berücksichtigt werden sollten. Auf allgemeingültige Anforderungen, die für jede Kategorie gleichermaßen zu berücksichtigen sind, soll dabei jedoch nicht ausführlicher eingegangen werden, da diese als vorausgesetzt betrachtet werden. So ist es unter anderem für jede der drei Kategorien wichtig, dass die Übertragung der Informationen in einem verteilten System gewisse Sicherheits-Standards bei der Übertragung (Verschlüsselung etc.) einhalten muss. Auch sollten grundlegende Design-Prinzipien wie beispielsweise die „Grundsätze der Dialoggestaltung“⁴⁷ oder Style-Guides gemäß dem eingesetzten Betriebssystem eingehalten werden. Die je nach Kategorie identifizierten Anforderungen lassen sich der folgenden Matrix entnehmen

⁴⁷ Quelle Grundsätze der Dialoggestaltung

Kommunikationskanal	Kategorie		
	1. Nutzer - direkt	2. Nutzer - indirekt	3. System
Senden (Initiator)	<ul style="list-style-type: none"> - Interaktionsmöglichkeit bereitstellen - Gebrauchstauglichkeit gewährleisten - Absenderinformationen ermitteln - Kontaktinformationen ermitteln - Kontextinformationen ermitteln - Informationsaufbereitung gewährleisten - Metainformationen ermitteln - Festlegung der Priorität gewährleisten 	<ul style="list-style-type: none"> - Aktivitäten des Nutzers ermitteln - Privatssphäre-Einstellung gewährleisten - Definition eigener Ereignisse - Ereignis-Typ ermitteln 	<ul style="list-style-type: none"> - Verkettete Ereignisse vermitteln - Ursprung ermitteln - Systemstatus ermitteln - Risiko für aktuellen Kontext ermitteln - Ereignis-Priorität ermitteln - Ggf. Maßnahmen zur Problembewältigung ermitteln
Empfangen (durch Nutzer)	<ul style="list-style-type: none"> - Absenderinformationen vermitteln - Kontaktinformationen vermitteln - Interaktionsmöglichkeit bereitstellen - Verfügbarkeit des Absenders vermitteln - Kontextinformationen vermitteln - Meta-Informationen ermitteln 	<ul style="list-style-type: none"> - Ereignis-Initiator vermitteln - Filtern relevanter Ereignisse ermöglichen - Klassifizierung der Ereignisse gemäß Ereignis-Typ - Aufdringlichkeit der Benachrichtigung minimieren 	<ul style="list-style-type: none"> - Ereignis-Hierarchie vermitteln - Ursprung vermitteln - Filtern relevanter Ereignisse - Systemstatus vermitteln - Priorität vermitteln - Interaktion für Anpassung des System-Status bereitstellen

Tabelle 3: Identifizierte Anforderungen an Ereignis-Benachrichtigungen in einem verteilten System

Die in dieser Matrix identifizierten Anforderungen basieren zwar auf einer ausführlichen Recherche von CSCW-Systemen sowie der verteilten Simulationsumgebung RCE, dennoch ist es durchaus möglich, dass nicht alle Anforderungen der jeweiligen Kategorien vollständig erhoben wurden. So können je nach Kontext des betrachteten CSCW-Systems noch weitere Anforderungen identifiziert werden, die in dieser Matrix nicht enthalten sind. Aus diesem Grund sollten das Modell und die entsprechende Matrix keinesfalls als vollständige Richtlinie dienen, nach denen sich Anforderungen an gebrauchstaugliche Benachrichtigungen erheben lassen, sondern ist derzeit eher als Orientierungshilfe zu bezeichnen.

5 Konzept

In diesem Kapitel erfolgt eine Betrachtung des erstellten Konzeptes für ein gebrauchstaugliches Benachrichtigungs-System der verteilten Simulationsumgebung RCE. Das Konzept basiert neben den identifizierten Anforderungen aus der zuvor durchgeführten Nutzungskontextanalyse im Wesentlichen auf den gewonnenen Erkenntnissen während der Analysephase die im Rahmen dieses Projektes erfolgte (siehe Kapitel 4). Das Konzept besteht im Wesentlichen aus folgenden Teilkonzepten:

- Tagging-System
- Notification-Center
- Notification-Bar
- Trigger-Events
- Tool-Awareness

Die einzelnen Teilkonzepte werden im Folgenden ausführlicher beschrieben und deren Bezug zu den in Kapitel 1.2 erläuterten Forschungsfragen sowie der relevanten Aspekte aus der Analysephase in Kapitel 4 verdeutlicht. An dieser Stelle sei zunächst anzumerken, dass die Entwicklung des Konzeptes einen stark iterativen Prozess bezeichnet, in dem verschiedene Gestaltungslösungen entworfen, weiterentwickelt oder auch wieder verworfen wurden. Dieser iterative Charakter der Konzeptphase, lässt sich innerhalb einer Ausarbeitung nur schwer erfassen. Die in diesem Kapitel vorgestellten Prototypen zu den jeweiligen Kategorien, basieren somit nicht zwingend auf einem einmaligen Entwurf, sondern wurden auf Grundlage eines iterativen Vorgehens stetig angepasst, um den Anforderungen an ein gebrauchstaugliches Benachrichtigungs-System gerecht zu werden. Besonders kritische Aspekte die eine iterative Überarbeitung erforderten, lassen sich Kapitel 6 dieser Ausarbeitung entnehmen.

5.1 Tagging-System

Bevor ausführlicher auf die weiteren Bestandteile des Konzeptes eingegangen wird, ist es zunächst notwendig, dass in diesem Projekt entworfene Tagging-System zu beschreiben. Das Konzept des Tagging-Systems stellt eine essentielle Funktionalität dar, die eine Grundlage für weitere Konzepte, wie dem Notification-Center und den Trigger-Events bildet. Es ermöglicht den Nutzern, verschiedene Artefakte⁴⁸ ihres

⁴⁸ Artefakte im Kontext von RCE können Workflows, Tools oder Ein- bzw. Ausgänge von Tools sein

aktuellen Aufgaben-Kontextes in dem RCE-System für eine effiziente Informations-Aufbereitung gemäß ihrer Anforderungen zu markieren und gewährleistet dadurch unter anderem ein hohes Maß an Individualisierbarkeit des Benachrichtigungs-Systems.

5.1.1 Tags

Das Wesentliche Element des Tagging-Systems bezeichnet die so genannten „Tags“⁴⁹. Ein Tag beschreibt dabei eine eindeutige Markierung, die der Nutzer beliebig auf Artefakte aus seiner RCE-Instanz anwenden kann. Zu diesen Artefakten zählen neben Workflows im Allgemeinen, auch einzelne Komponenten oder spezifische Ein-/Ausgänge dieser. Die Tags können durch den Nutzer individuell definiert und angepasst werden. Um eine Differenzierung zwischen mehreren Tags zu gewährleisten, wird ein Tag stets durch einen eindeutigen Namen, sowie eine farbliche Darstellung definiert (siehe Abb. x.xx). Ein erstelltes Tag-Element kann dabei beliebig oft auf verschiedene Artefakte angewandt werden, und ermöglicht es den Nutzern, ihre Artefakte in eigenen Tag-Kategorien zusammen zu fassen.



Abbildung 21: Gestaltungslösung der Tag-Darstellung mit eindeutiger Bezeichnung und farblicher Differenzierung

5.1.2 Tag-Interaktion

Um den Nutzern einen Einsatz von Tags zu ermöglichen, sieht das Konzept verschiedene Interaktions-Möglichkeiten vor. Diese Interaktionen stellen Funktionalitäten des Tagging-Systems bereit, die im Folgenden ausführlicher erläutert werden.

⁴⁹ dt.: Markierung, Kennzeichnung

5.1.2.1 Tags anlegen

Das Tagging-System ermöglicht die Erstellung eigener Tags durch die Nutzer, die anschließend zur Markierung von Artefakten des RCE-Systems eingesetzt werden können. Der Nutzer kann die Funktionalität zum Anlegen eines Tags einerseits über das Kontextmenü des jeweiligen Artefaktes erreichen oder über die Notification-Einstellungen vornehmen. Wird das Anlegen eines neuen Tags über das Kontext-Menü gewählt, so erfolgt die Darstellung eines Dialoges, in dem alle relevanten Einstellungen vorgenommen werden können. Wird der Dialog z.B. über das Kontext-Menü einer Komponente aufgerufen, so kann diese Komponente direkt mit dem neu erstellten Tag markiert werden. Diese Art der Interaktion wird nachfolgenden Abbildung verdeutlicht.

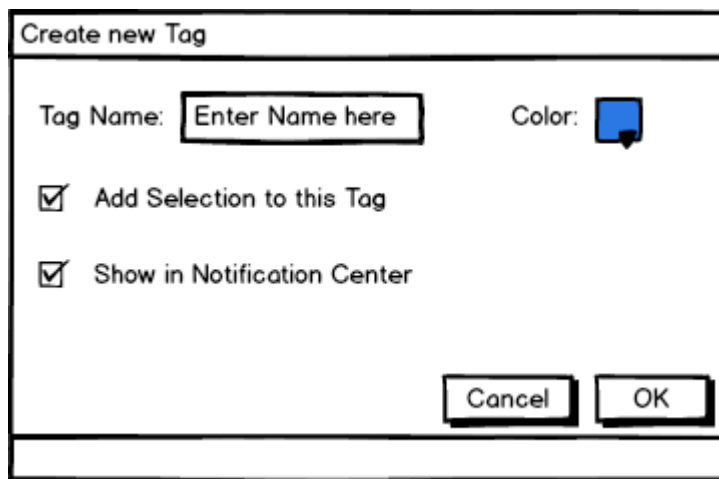


Abbildung 22: Gestaltungslösung eines Dialoges zur Erstellung eines neuen Tags

5.1.2.2 Tags bearbeiten/löschen

Über die Einstellungen der Notifications, lassen sich neben dem Anlegen neuer Tags auch erstellte Tags anpassen oder löschen. Die Bearbeitung von Tags über die Einstellungen soll gewährleisten, dass der Nutzer, je nach Änderung des Kontextes, seine Tags gemäß der Anforderungen anpassen kann, ohne diese löschen und neu anlegen zu müssen. So können neben dem Namen und der farblichen Differenzierung auch die markierten Artefakte angepasst werden.

5.1.2.3 Tags anwenden

Das Markieren von Artefakten durch angelegte Tags, soll dem Nutzer durch einen möglichst geringen Aufwand zugänglich sein, um ihn bei seiner primären Aufgabenbewältigung nicht nachhaltig zu beeinträchtigen. Die erstellten Tags lassen

sich daher über das Kontext-Menü des jeweiligen Artefaktes auswählen (siehe Abb. 22). Artefakte die mit einem Tag markiert wurden, sind durch eine entsprechende farbliche Differenzierung in dem Editor des RCE-System hervorgehoben.

Die weitere Verwendung des Tagging-Systems wird insbesondere in den Kapitel x.xx. sowie y.yy. verdeutlicht.

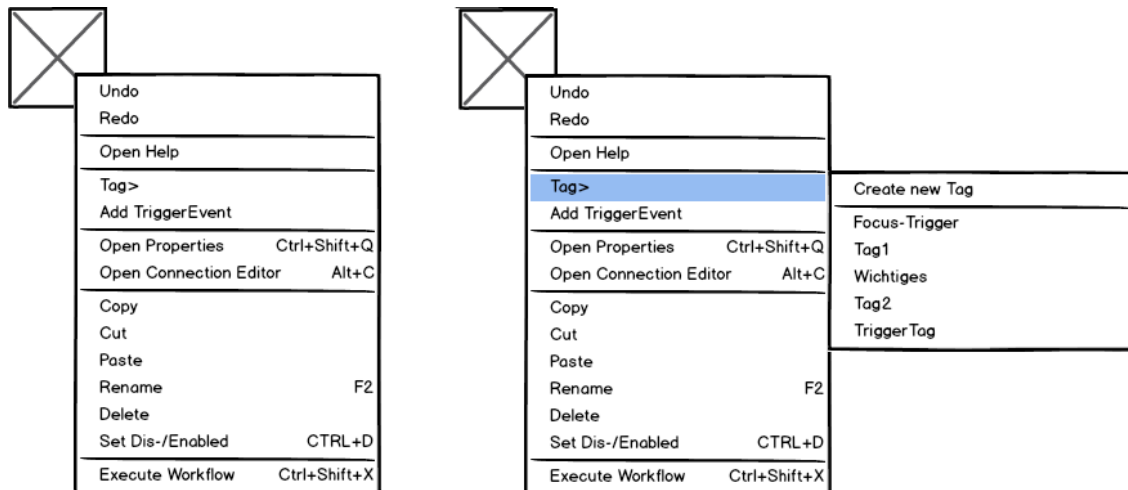


Abbildung 23: Kontextmenü mit Einträgen des Tagging-Systems. Bereits erstellte Tags lassen sich anhand der Listendarstellung (rechts) einem Artefakt zuordnen.

5.2 Notification Center

In der verteilten Simulationsumgebung RCE treten eine Vielzahl an Ereignissen ein, die dem Nutzer des Systems je nach Kontext seiner aktuellen Aufgabe auf eine gebrauchstaugliche Weise vermittelt werden müssen. Die Informationen über ein Ereignis können dabei potenziell über verschiedene Informationsträger, gemäß der in Kapitel 4.5. aufgeführten Darstellungsformen erfolgen. Dem Nutzer werden in RCE Ereignisse bislang unter anderem durch Popup-Dialoge (z.B. der Verbindungsabbruch zu einem Server/Netzwerk), verschiedene Views (z.B. Workflow List, Workflow Console, etc.) oder durch extern gesicherte Log-Dateien vermittelt. Aus der Analysephase in Kapitel 4, in der unter anderem die Wechselwirkung aus Aufmerksamkeit des Menschen und Nutzen eines Benachrichtigungs-Systems behandelt wurden, wird deutlich, dass die aktuelle Vielfalt an möglichen Informationsquellen keine effiziente Vermittlung von Ereignissen zur menschlichen Verarbeitung ermöglicht. Möchte ein Nutzer bestimmte Ereignisse innerhalb des bestehenden RCE-Systems nachvollziehen können, so muss dieser entsprechende Informationen bislang über verschiedene Ansichten der Benutzeroberfläche finden. Die benötigte Aufmerksamkeit, um Informationen über ein Benachrichtigungs-System zu entnehmen,

soll jedoch nach McCrickard et.al. möglichst gering gehalten werden, um ein angemessenes Verhältnis aus Aufmerksamkeits-Kosten und Benachrichtigungs-Nutzen zu gewährleisten, und somit ein hohes Maß an Effizienz zu erreichen. Die Vielfalt der Informations-Quellen steht daher in einem Widerspruch zu dem Modell aus Aufmerksamkeit und Nutzen nach McCrickard et.al., wenn ein Nutzer zur Erhebung von relevanten Informationen verschiedene Informationsquellen beachten muss.

Diese Grundlagen bilden das Konzept des „Notification-Centers“, in dem alle für den Nutzer relevanten Informationen zentral dargestellt werden, ohne dass dieser weitere Quellen betrachten muss. Das Notification-Center ist unter anderem über die Notification-Bar erreichbar und kann, entsprechend der Anforderung „Individualisierung [DIN EN ISO 9241 Teil 110], je nach Kontext des Nutzer beliebig angepasst werden.

Das Ziel, eine zentrale „Anlaufstelle“ an Ereignis-Informationen für den Nutzer zu schaffen, birgt allerdings auch einige Risiken, die bei der Konzeption zu berücksichtigen sind. So besteht ein Schwerpunkt des Notification-Center-Konzeptes darin, die teils enorme Anzahl eingehender Ereignis-Benachrichtigungen so darzustellen, dass der Nutzer die für ihn relevanten Informationen dennoch effizient wahrnehmen kann. Im Optimalfall soll der Nutzer darüber hinaus nicht selber nach Kontext-relevanten Informationen suchen müssen, sondern diese, je nach Relevanz, durch das System dem Nutzer direkt bereitgestellt werden. Um diesen Ansprüchen gerecht zu werden, besteht das Konzept des Notification-Centers aus den Komponenten der System-Notifications, der Tool-Notifications und der Workflow-Notifications, die im Folgenden näher erläutert werden.

5.2.1 Notification-Groups

Wie in Kapitel 2.3. erläutert, treten in RCE Ereignisse auf, die sich auf verschiedene Aspekte lokaler sowie verteilter Aktivitäten des Systems beziehen können. Diese Aspekte sind für die Nutzer des RCE-Systems jedoch nicht zu jedem Zeitpunkt alle gleichermaßen relevant und stehen in direkter Abhängigkeit des Nutzungskontextes, in dem sich der Nutzer bewegt. Erfolgt nun die Darstellung aller eingehenden Ereignisse über eine zentrale Informationsquelle wie dem Notification-Center, kann das Filtern der für den Nutzer relevanten Informationen mit einem erheblichen Aufwand verbunden sein. Das Konzept des Notification-Centers sieht daher die Klassifizierung eingehender Ereignis-Benachrichtigungen in verschiedene Kategorien vor, die so genannten „Notification-Groups“. Diese sollen dem Nutzer eine effiziente Verarbeitung relevanter Informationen ermöglichen. Die Anforderung an eine Klassifizierung eingehender Benachrichtigungen lässt sich zudem dem in der Analysephase vorgestellten Modell

(siehe Kapitel 4) entnehmen. Die Klassifizierung erfolgte im Rahmen dieses Konzeptes auf Grundlage einer Identifikation aller relevanter Ereignisse, welche anschließend gemäß der KJ-Methode zu angemessenen Kategorien zusammengefasst wurden. Folgende Kategorien konnten dabei identifiziert werden und sind in dem Notification-Center nativ enthalten:

System-Notifications

Bei System-Notifications handelt es sich um eingehende Ereignisse, die nicht unmittelbar durch einen Nutzer ausgelöst werden und eine Änderung des System-Status sowie allgemeine Hilfestellungen für den Nutzer beschreiben. Die Änderung des System-Status kann sich dabei je nach Kontext eines Nutzers nachhaltig auf dessen Aufgabenbewältigung auswirken, weshalb entsprechende Benachrichtigungen als eigene Kategorie in dem Notification-Center enthalten sind. Zu System-Notifications zählen unter anderem Informationen über die Netzwerkkonnektivität anderer RCE-Instanzen (wie Servern oder Clients anderer Nutzer), sowie Informationen über die Datenverarbeitung des Systems (z.B. speichern von Workflows, laden der Konfigurationsdatei, exportieren von Informationen, etc.).

Tool-Notifications

Die Kategorie der Tool-Notifications beschreibt alle eingehenden Ereignisse, die sich auf die in RCE eingesetzten Komponenten eines Nutzers beziehen. Neben Informationen über den Status verschiedener Komponenten zur Laufzeit (z.B. idle, running oder finished) gehört auch die Verfügbarkeit verteilt eingesetzter Komponenten. Da der Einsatz von Komponenten ein essentieller Bestandteil von Workflows in RCE ist und je nach Vielfalt der eingesetzten Komponenten eine Vielzahl an Komponenten-Ereignissen eintreten, werden diese in dem Notification-Center durch die Kategorie der Tool-Notifications zusammengefasst.

Workflow-Notifications

Benachrichtigungen über eintretende Workflow-Ereignisse in RCE werden über das Notification-Center in der Kategorie der Workflow-Notifications dargestellt. Workflow-Notifications vermitteln beispielsweise Informationen über die Zustände laufender Workflows (z.B. started, paused, finished) sowie über Abbrüche dieser.

Die soeben definierten Kategorien des Notification-Centers, stellen jedoch keine allgemeingültige Unterteilung dar, die in jedem Kontext gleichermaßen zu einer effizienten Ereignis-Vermittlung führt. Je nach Arbeitsweise eines Nutzers, seiner

Aufgaben und dem Nutzungskontext im Allgemeinen, ist eine solche Unterteilung unangemessen und muss in deren Granularität an die Anforderungen des aktuellen Kontextes angepasst werden. So kann es unter anderem vorkommen, dass ein Nutzer sich nur Informationen über seine eigens entwickelten Tools in dem Notification-Center anzeigen lassen möchte oder das er keine Informationen über System-Benachrichtigungen erhalten will. Diese Annahme führt dazu, dass die soeben vorgeschlagenen Kategorien des Notification-Centers lediglich als „Standard“-Kategorien zu betrachten sind und eine Anpassung in weitere Notification-Groups durch den Nutzer gewährleistet werden muss. Das Konzept der Notification-Groups beschreibt daher einen Ansatz, der in hohem Maße individualisierbar sein soll. Die Individualisierbarkeit des Notification-Centers lässt sich dem Kapitel x.xx. dieses Kapitels entnehmen.

5.2.2 Syntax einer Ereignis-Benachrichtigung

Für eine effiziente Vermittlung von Ereignis-Benachrichtigungen innerhalb des Notification-Centers ist es grundsätzlich erforderlich, eine angemessene Syntax zu finden, die sich gleichermaßen für alle Kategorien an eingehenden Ereignissen anbietet. Eine einheitliche Syntax, die Ereignis-übergreifend Informationen darstellt, ermöglicht es unter anderem den Aspekt des „ease of learning“ der Gebrauchstauglichkeit nach der DIN EN ISO 9241 Teil 110 zu erfüllen. Durch eine einheitliche Darstellung wird so gewährleistet, dass die Nutzer des Systems die dargestellten Informationen kognitiv effizient verarbeiten können. Die als relevant erachteten Bestandteile der Syntax basieren auf einer Betrachtung der bereits in RCE dargestellten Informationen und unter Berücksichtigung verschiedener Anforderungen an die Benachrichtigung nach dem in Kapitel 4. vorgestellten Modell. Die Syntax einer Ereignis-Benachrichtigung, wie sie im Notification-Center eingesetzt wird, setzt sich daher aus folgenden Komponenten zusammen:

Indikator

Der Indikator dient dem Nutzer als eine grafische Repräsentation zur effizienten Klassifikation eines Ereignisses. Je nach Ereignis kann der Nutzer bereits anhand des Indikators interpretieren, ob dieses in seinem aktuellen Kontext von Bedeutung ist. Der Einsatz von Indikatoren wird bereits durch einige Aspekte des RCE-Systems abgedeckt (siehe Kapitel 3.2.3.)

Nachricht

Diese Komponente definiert den eigentlichen Inhalt einer Benachrichtigung und enthält

je nach Ereignis die entsprechenden Informationen in einer, dem Menschen verständlichen Fachsprache.

Instanz

Da in der verteilten Simulationsumgebung RCE Ereignisse sowohl lokal als auch verteilt auftreten können, ist es notwendig, Ereignis-Benachrichtigungen den entsprechenden Instanzen zuordnen zu können, von denen diese Ereignis-Benachrichtigung ausgeht. Die Syntax einer Ereignis-Benachrichtigung sieht daher auch eine Vermittlung der Ereignis-Instanz vor.

Zeitstempel

Der Zeitstempel definiert, zu welchem Zeitpunkt das dargestellte Ereignis an der entsprechenden Instanz eingetreten ist.

Meta-Informationen

Die Komponente der Meta-Informationen bezeichnet optionale Informationen, die je nach Ereignis relevant sein können. Da sich nicht alle Kategorien an Ereignissen gleichermaßen durch die zuvor erwähnten Komponenten hinreichend beschreiben lassen, wird diese zusätzliche Komponente in die Syntax aufgenommen, um beispielsweise Ereignis-spezifische Informationen darzustellen. So kann über die Meta-Informationen beispielsweise abgebildet werden, wann ein nicht verfügbares Tool wieder zur Verfügung gestellt wird.



Abbildung 24: Gestaltungslösung der Benachrichtigungs-Syntax mit grafischen Indikatoren

5.2.3 Notification-Fokus

Eine besondere Herausforderung bei der Erstellung eines gebrauchstauglichen Benachrichtigungs-System lag darin, aus der Vielzahl an eingehenden Ereignis-Informationen, die dem Nutzer relevanten Informationen hervorzuheben. Dadurch soll dieser bei seiner Aufgabenbewältigung unterstützt werden, ohne dass er selber aktiv

nach Informationen suchen muss. Zwar sieht das Notification-Center eine Klassifizierung der Benachrichtigungen vor, anhand derer der Nutzer effizienter Informationen wahrnehmen kann, dennoch kann auch hier trotz einer Berücksichtigung verschiedener Aspekte der Gebrauchstauglichkeit nicht sichergestellt werden, dass das Finden relevanter Informationen ohne einen erheblichen Aufwand realisierbar ist.

Das Konzept des Notification-Fokus als ein Bestandteil des Notification-Centers definiert daher einen Bereich auf dem Interface, in dem nur kontextsensitive Informationen dargestellt werden sollen. Die Definition, welche Informationen in dem Notification-Fokus als relevante Kontext-Informationen eines Nutzers hervorzuheben sind, ist dabei jedoch nicht trivial vorzunehmen. So stellt sich zunächst die Frage, was den aktuellen Kontext eines Nutzers bei der Bewältigung einer Aufgabe mit RCE umfasst. Zunächst wurde die Annahme getroffen, dass sich der aktuelle Aufgaben-Kontext stets aus den, durch den Nutzer erstellten sowie ausgeführten, Workflows innerhalb der Simulationsumgebung ergibt. Nach einer Bewertung erster Gestaltungslösungen mit Mitgliedern des RCE-Entwicklerteams wurde allerdings deutlich, dass diese Annahme den Kontext eines Nutzers nicht ganzheitlich beschreiben kann, und der Kontext je nach Szenario des Nutzers variiert. So besteht die Möglichkeit, dass sich ein Nutzer auf Informationen eines verteilt durchführenden Workflows fokussieren möchte, während er selbst über den Editor der RCE-Benutzeroberfläche einen eigenen Workflow bearbeitet. Legt das Benachrichtigungs-System nun den Fokus selbstständig auf den Workflow des Nutzers, so erhält dieser keine Informationen über den für ihn relevanten Workflow innerhalb des Notification-Fokus. Dieses Szenario verdeutlicht, dass das System allein nicht die Entscheidung vornehmen sollte, wann welche Informationen im besonderen Interesse des Nutzers liegen. Da der Fokus eines Nutzers je nach Aufgabe und persönlicher Arbeitsweise variieren kann, sieht das Konzept eine individuelle Anpassung des Notification-Fokus vor, in dem der Nutzer selber definieren kann, worauf er fokussieren möchte. Dem Nutzer stehen dabei folgende Kategorien an Informationen zur Auswahl, die innerhalb des Notification-Fokus darstellbar sind:

Current Workflows

Die Informationen über alle, durch den Nutzer gestartete oder geöffnete Workflows, lassen sich über die Kategorie der „Current Workflows“ in dem Notification-Fokus anzeigen. Sofern der Nutzer diese Kategorie nicht bewusst in den Einstellungen des Notification-Centers deaktiviert hat, ist diese Kategorie als ein Standardwert des Notification-Fokus aktiviert.

Error Messages

Auftretende Fehlermeldungen, unter anderem bedingt durch fehlerhafte Komponenten, oder allgemeinen Systemfehlern, werden durch die Kategorie der „Error Messages“ im Notification-Fokus abgebildet. Ähnlich der Current Workflows, ist auch diese Kategorie als Standard des Notification-Fokus gesetzt, lässt sich jedoch durch den Benutzer je nach Anforderung deaktivieren. In dem Notification-Fokus soll eine ganzheitliche Betrachtung auftretender Fehlermeldungen erfolgen, weshalb sowohl Meldungen aus der Log View (Errors) als auch aus der Workflow-Console View über den Notification-Fokus vermittelt werden.

„Fokus“-Tag

Der Einsatz des „Fokus“-Tags als eigene Kategorie des Notification-Fokus definiert eine Methode, die ein hohes Maß an Individualisierbarkeit realisiert. Durch die Verwendung eines vordefinierten Fokus-Tags innerhalb des Tagging-Systems (siehe Abschnitt 5.1.), kann der Nutzer Artefakte des RCE-Systems markieren und diese dadurch in den Notification-Fokus aufnehmen. Das Markieren der Artefakte mit dem Fokus-Tag kann dabei gemäß der Interaktions-Möglichkeiten des Tagging-Systems über ein Kontextmenü erfolgen und ermöglicht dadurch dem Nutzer, eine effiziente Anpassung des Notification-Fokus.

User Messages

Das Konzept dieser Ausarbeitung sieht auch eine direkte Kommunikation zwischen Nutzern vor, wie dies in klassischen CSCW-Systemen (siehe Kapitel x.xx.) realisiert wird. Entsprechende Ereignisse, die unmittelbar von Nutzern ausgehen und an einen anderen Nutzer oder eine Gruppe gerichtet sind, werden zwar in diesem Konzept nicht berücksichtigt, die Anforderungen an eine direkte Kommunikation zwischen Nutzern konnte jedoch unter anderem im Prozess der Nutzungskontextanalyse erhoben werden. Da eingehende Ereignis-Benachrichtigungen anderer Nutzer bewusst durch die verteilte Instanz an den Nutzer übermittelt werden, ist allgemein davon auszugehen, dass diese Informationen eine hohe Priorität besitzen, weshalb Nutzer-Benachrichtigungen als Standard-Wert für den Notification-Fokus aktiviert sind.

Der Notification-Fokus wird im oberen Bereich des Notification-Centers dargestellt (siehe Abb. 26) und wird farblich hervorgehoben. Anders als die Notification-Groups, kann der Notification-Fokus nicht auf der Benutzeroberfläche des Notification-Centers verschoben werden, lässt sich jedoch über die Einstellungen durch den Nutzer deaktivieren und in dessen Inhalt anpassen.

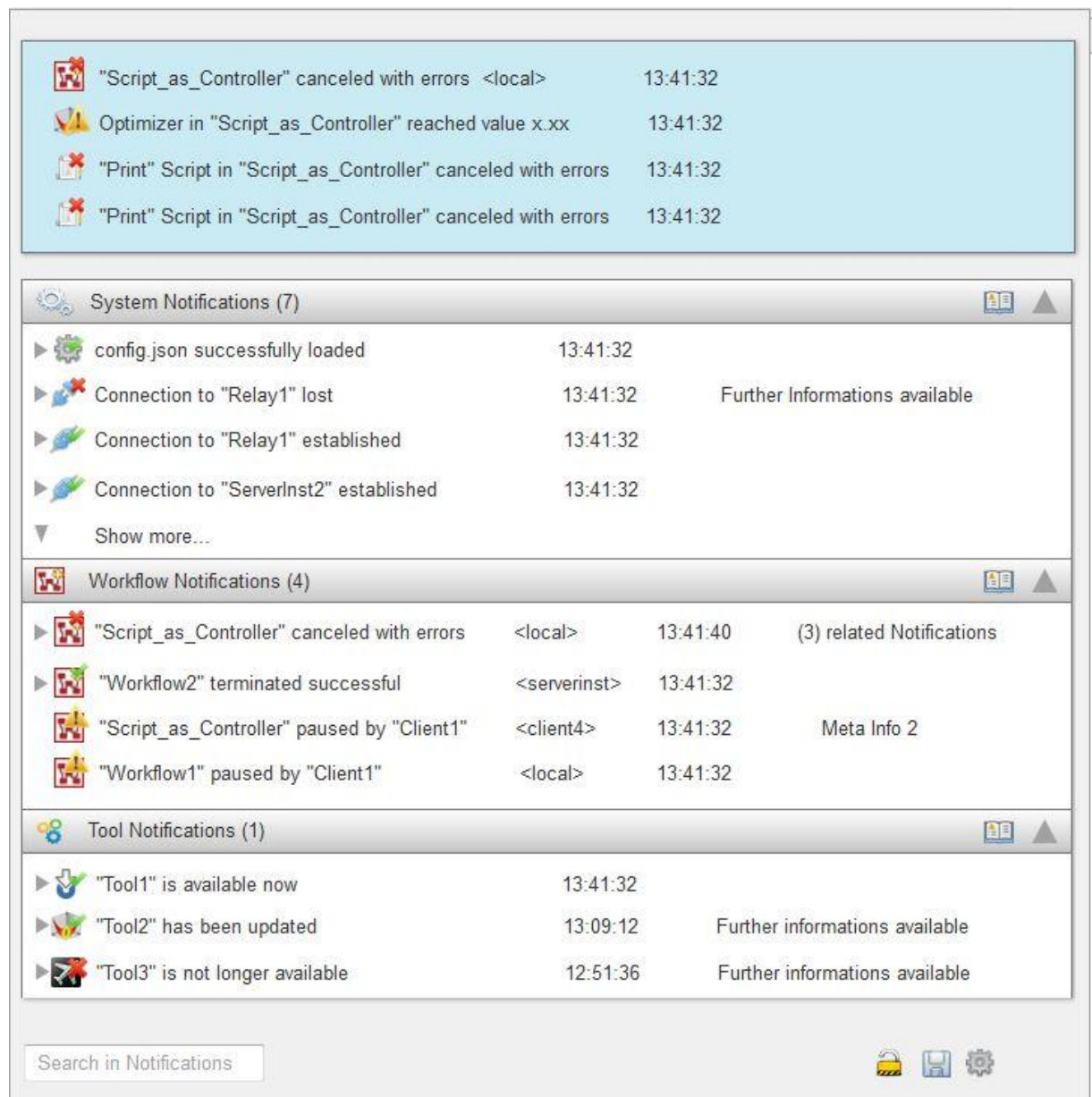


Abbildung 25: Gestaltungslösung des Notification-Centers mit Notification-Fokus im oberen Bereich.

5.2.4 Notification-Cluster

Viele der in RCE auftretenden Ereignisse sowie der daraus resultierenden Benachrichtigungen, stehen in einem unmittelbaren Zusammenhang zu anderen Ereignissen. Ein besonderes Risiko bei der Erstellung eines Konzeptes für das Benachrichtigungs-System lag zudem darin, dass die Übersichtlichkeit des Notification-Centers aufgrund der teils hohen Anzahl eingehender Benachrichtigungen nur schwer gewährleistet werden kann. Diese Betrachtung führte zu einer Konzeption der „Notification-Cluster“, in denen eine Zusammenfassung eingehender

Benachrichtigungen gemäß einer angemessenen Logik erfolgt. Durch das Konzept des Notification-Clusters, wird eine Reduktion der dargestellten Ereignis-Benachrichtigungen vorgenommen, mit dem Ziel, die Übersichtlichkeit des Notification-Centers auch bei einer Vielzahl an Informationen zu erhalten. Das Konzept sieht dabei zwei Arten logischer Zusammenhänge vor:

Kausale Zusammenfassung

Ereignisse die nicht unmittelbar durch den Nutzer ausgelöst wurden, sollten diesem hinreichende Informationen zur Verfügung stellen, anhand derer der Nutzer den Auslöser dieses Ereignisses identifizieren kann. Diese Anforderung konnte unter anderem auch in dem in Kapitel 4. vorgestellten Modell identifiziert werden. Oftmals findet in solchen Szenarien eine Verkettung von Ereignissen statt, die sich bis zu ihrem Ursprung zurückverfolgen lassen. So treten auch in der verteilten Simulationsumgebung Ereignisse ein, die sich durch einen kausalen Zusammenhang verknüpfen lassen. Auf Grundlage dieser Betrachtung, sieht das Konzept eine Zusammenfassung von kausalen Ereignissen vor. Ein typisches Szenario für einen kausalen Notification-Cluster ist unter anderem der Abbruch eines Workflows, aufgrund eines nicht mehr verfügbaren verteilten Tools, da die entsprechende Server-Instanz geschlossen wurde. Eine mögliche Übertragung dieses Szenarios in einen Notification-Cluster kann der Abb. 27 entnommen werden.

Zeitliche Zusammenfassung

In dem RCE-System treten zudem Ereignisse auf, die sich auf eine bestimmte Kategorie beziehen und innerhalb eines kurzen Zeitfensters aufeinander folgen. Da viele dieser Ereignisse für den Nutzer nicht zwingend relevant sind, würde eine separate Darstellung jedes dieser Ereignisse einer Kategorie ggf. Kontext-relevante Informationen verdecken. Das Konzept des Notification-Clusters, sieht daher auch eine zeitliche Zusammenfassung von Ereignissen einer Kategorie vor. So lassen sich beispielsweise schnell aufeinander folgende Benachrichtigungen über Verbindungsabbrüche in einen solchen Notification-Cluster zusammenfassen, den der Nutzer je nach Bedarf ausführlicher betrachten kann.



Abbildung 26: Gestaltungslösung geschlossener Notification-Cluster



Abbildung 27: Gestaltungslösung eines geöffneten Notification-Clusters. Über die nebenstehenden Pfeile, lässt sich ein Notification-Cluster öffnen bzw. schließen

Die soeben vorgestellten Möglichkeiten, Ereignis-Benachrichtigungen zusammenzufassen, sollte jedoch nicht als ganzheitlich betrachtet werden. So ist nicht auszuschließen, dass sich je nach Anforderungen an das Benachrichtigungs-System auch weitere Ereignisse zu Notification-Clustern zusammenfassen lassen. Die Darstellung der Notification Cluster innerhalb des Notification-Centers orientiert sich an dem Aufbau einer Baumstruktur, und ermöglicht den Nutzern eine effiziente Interpretation der chronologischen Zusammenhänge. Die Darstellung eines Notification-Clusters, hebt sich durch entsprechende Meta-Informationen der Benachrichtigungs-Syntax von einzelnen Ereignis-Benachrichtigungen ab. Verdeutlicht wird dies unter anderem durch die in Abbildung 27. vorgeschlagene Gestaltungslösung, die das Konzept des Notification-Centers darstellt.

5.2.5 Individualisierbarkeit

Die Anforderungen an die Darstellung von Informationen innerhalb des Benachrichtigungs-Systems, können je nach Nutzungskontext stark variieren. Wie in den einzelnen Bestandteilen dieses Konzeptes angedeutet, ist eine allgemeingültige Aussage darüber, welche Informationen in welchem Kontext relevant sind, nicht annehmbar. Dementsprechend ist eine effiziente Wahrnehmung eingehender Ereignis-Benachrichtigungen durch den Nutzer nicht möglich, wenn ausschließlich das Benachrichtigungs-System vorgibt, ob und wie eingehende Benachrichtigungen dargestellt werden. Die Anforderung der Individualisierbarkeit des Systems wurde zu dem auch im Rahmen der durchgeführten Nutzungskontextanalyse als wichtig erachtet. Das Konzept des Notification-Centers sieht daher eine individuelle Anpassung der Benutzeroberfläche sowie der darin enthaltenen Informationen vor. Die Anpassung des

Notification-Center kann sowohl über den Einstellungs-Dialog des RCE-Systems als auch über die Benutzeroberfläche des Notification-Centers erfolgen und ermöglicht verschiedene Anpassungen.

5.2.5.1 Notification-Fokus

Wie in Kapitel 5.2.3. erläutert, beinhaltet der Notification-Fokus besonders kritische Informationen, die in einem direkten Bezug zu dem Kontext eines Nutzers stehen sollen. Zwar definiert das Benachrichtigungs-System Standard-Werte für diesen Bereich des Notification-Centers (siehe Kapitel 5.2.1.), jedoch können diese je nach Anforderungen eines Nutzers keine angemessene Darstellung von Informationen gewährleisten. Der Notification-Fokus sieht daher neben einer Anpassung durch den in Kapitel 5.2.3. beschriebenen Fokus-Tag eine umfangreiche Individualisierbarkeit der dargestellten Informationen vor. Der Nutzer kann dabei individuell entscheiden, welche der Notification-Fokus-Kategorien er anzeigen möchte, und darüber hinaus die einzelnen Kategorien durch zusätzliche Filter anpassen. Der Notification-Fokus lässt sich zudem, ähnlich der einzelnen Notification-Groups innerhalb des Notification-Centers, deaktivieren. Durch die Voreinstellung der Standard-Werte, sowie der umfangreichen Anpassungsfähigkeit des Benachrichtigungs-Systems, sollen sowohl unerfahrene als auch erfahrene Nutzer das System gleichermaßen effizient nutzen können und gemäß ihrer Anforderungen anpassen.

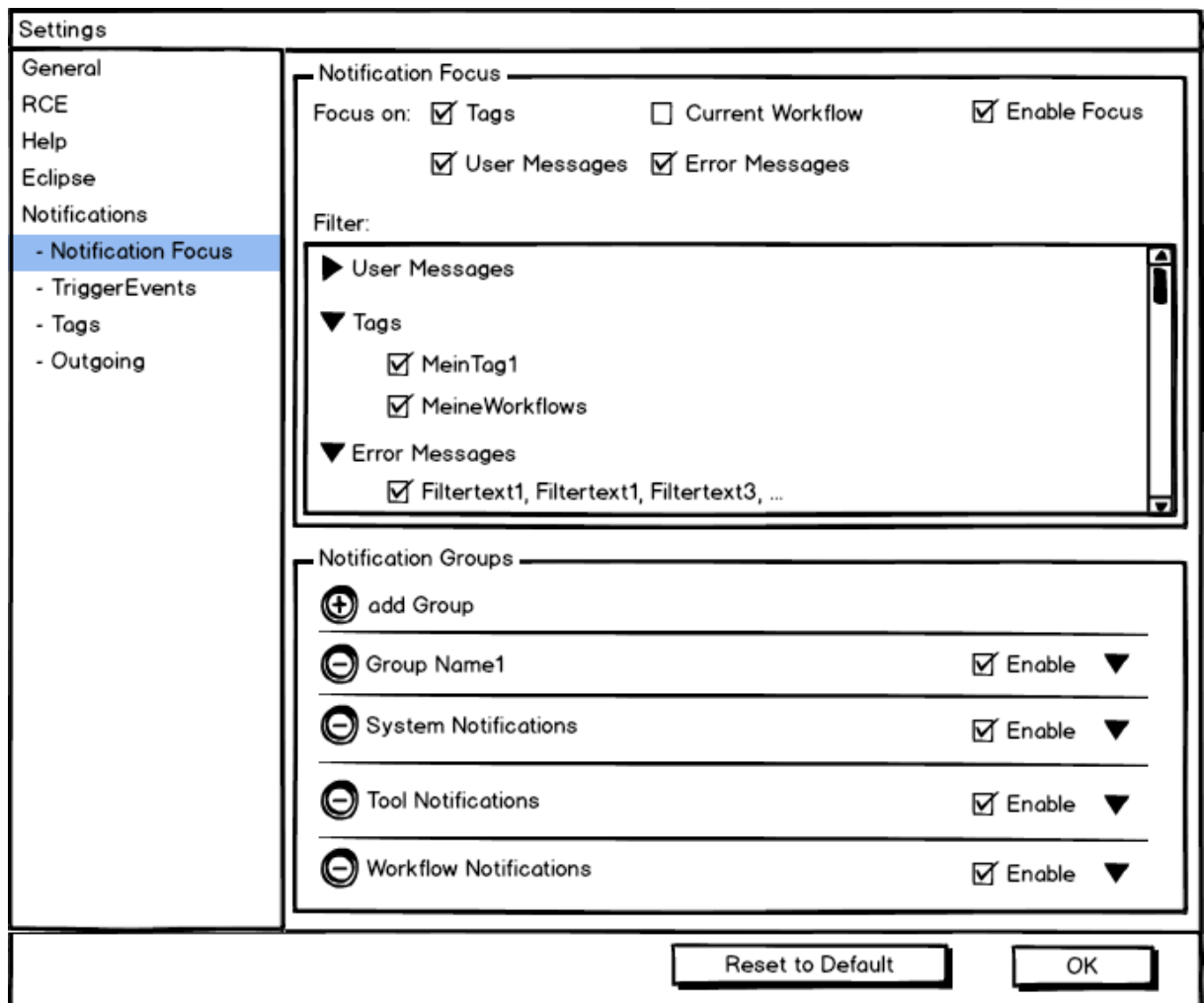


Abbildung 28: Einstellungen des Notification-Centers anhand eines Prototypens mit Anpassungsmöglichkeiten des Notification-Fokus sowie der Notification-Groups.

Die Individualisierbarkeit des Notification-Fokus erfolgt über die allgemeinen Einstellungen des Notification-Centers und wird, für eine bessere optische Differenzierung, in einem eigenständigen Bereich dargestellt (siehe Abb. 28). Die Anordnung der Einstellungen des Notification-Centers orientiert sich dabei an der Anordnung der tatsächlichen Benutzeroberfläche und soll dadurch ein hohes Maß an Selbstbeschreibungsfähigkeit, gemäß der DIN EN ISO 9241 Teil 110 gewährleisten. Je nach Auswahl der darzustellenden Kategorien im oberen Bereich der Fokus-Einstellungen, werden diese in der darunter liegenden Tabelle angezeigt und ermöglichen eine zusätzliche Filterung innerhalb der einzelnen Kategorien. Die möglichen Filter-Optionen hängen dabei von den jeweiligen Kategorien der Benachrichtigungen ab. Möchte ein Nutzer seine, mit dem „Fokus-Tag“ markierten Artefakte anpassen, so stellt ihm das Interface eine Darstellung der markierten Artefakte zur Verfügung, die dann aktiviert bzw. deaktiviert werden können. Sollen hingegen

Fehlermeldungen gefiltert werden, so kann der Nutzer hingegen verschiedene Filterbegriffe, getrennt durch Kommata, selber definieren. So ist es unter anderem möglich, dass ein Nutzer sich nur Error-Messages anzeigen lässt, welche sich auf ein bestimmtes Tool beziehen.

5.2.5.2 Notification-Groups

Die Individualisierung der Notification-Groups, kann über die allgemeinen Einstellungen des Notification-Centers erfolgen. Bei der Anpassung der Notification-Groups, kommt insbesondere das Konzept des Tagging-Systems zum Einsatz. So können aus den, durch den Nutzer selbst definierten Tags, eigene Notification-Groups erstellt und in dem Notification-Center angezeigt werden.

Erstellt ein Nutzer eine neue Notification Group über die Einstellungen des Notification-Centers, wie der Gestaltungslösung in Abbildung 29 zu entnehmen, so kann dieser zunächst eine Kategorie von Ereignis-Benachrichtigungen wählen und die daraus resultierenden Informationen zusätzlich in der neu erstellten Notification-Group filtern. Neben den vordefinierten Notification-Groups, stehen dem Nutzer dabei auch seine eigens definierten Tags als Kategorien für eine Notification-Group zur Verfügung. Möchte der Nutzer innerhalb des Notification-Centers beispielsweise Informationen über alle seine entwickelten Komponenten erhalten, so kann er diese über das Tagging-System durch einen angemessenen Tag markieren, um diesen Tag anschließend als eigene

Notification-Group im Notification-Center darzustellen. Die selbst definierten Notification-Groups können zudem durch den Nutzer benannt werden, um diese in dem Notification-Center schnell identifizieren zu können. Das Hinzufügen bzw. Entfernen von Notification-Groups, wird in den Einstellungen metaphorisch durch entsprechende „+“ bzw. „-“, Symbole dargestellt. Diese Darstellung orientiert sich ebenfalls an den Aspekt der Selbstbeschreibungsfähigkeit, gemäß der DIN EN ISO 9241 Teil 110.

Eigens definierte Notification-Groups, können zudem nicht nur eine Kategorie enthalten, sondern lassen sich beliebig um weitere Kategorien ergänzen. Dies wird in dem Konzept durch die „add Category“-Funktionalität gewährleistet. Durch dieses Konzept kann der Nutzer sich innerhalb einer Notification-Group z.B. alle Informationen über seine eigenen Komponenten, als auch Informationen über verschiedene System-Ereignisse anzeigen lassen.

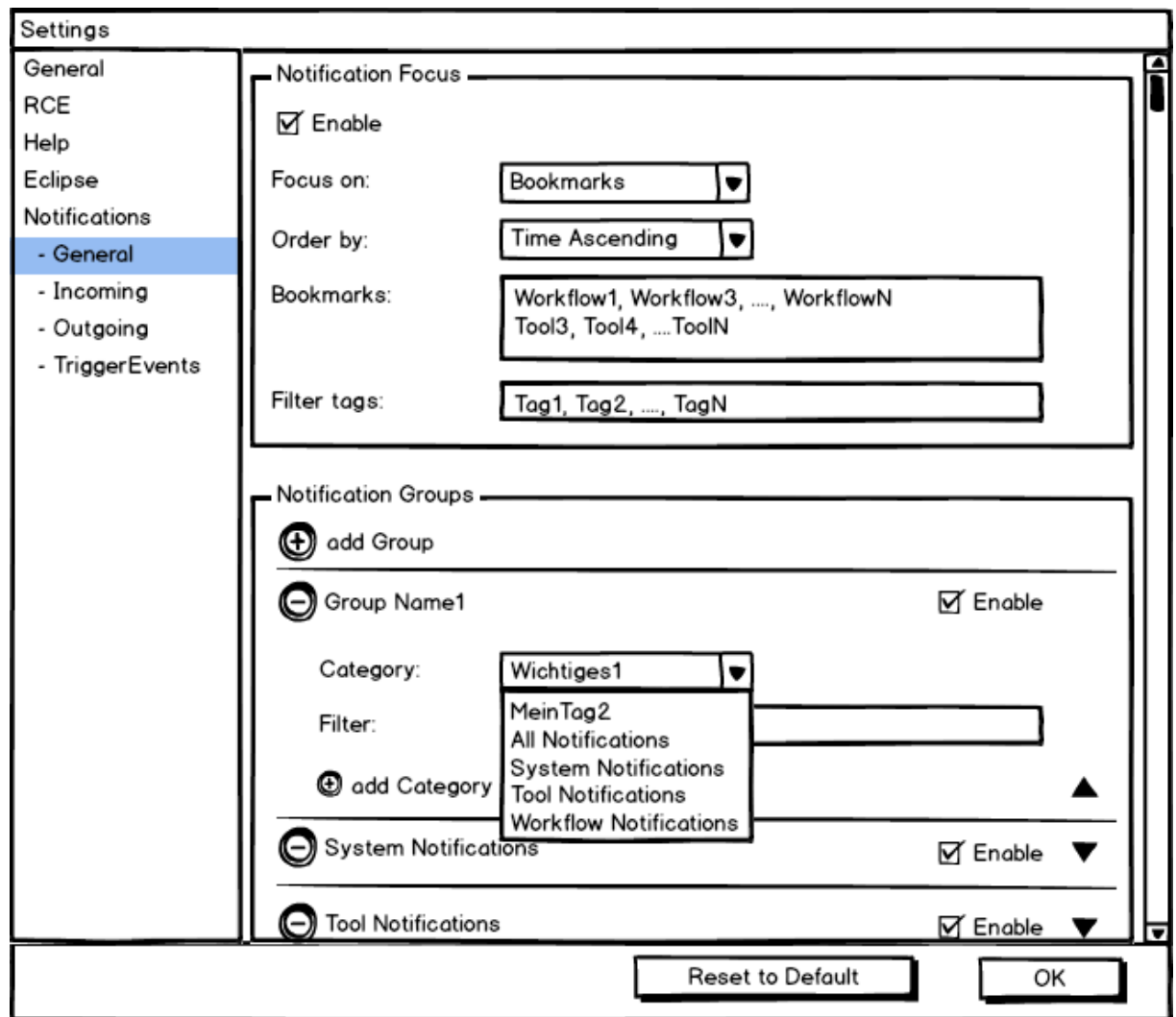


Abbildung 29: Gestaltungslösung für das Anlegen einer neuen Notification-Group.

Welche der Notification-Groups in dem Notification-Center dargestellt werden, lässt sich über die entsprechenden Checkboxes der Notification-Groups definieren. Die Anordnung der Groups ist anschließend über die Benutzeroberfläche des Notification-Centers anpassbar. Die Anordnung erfolgt dabei durch ein Drag-und Drop-Paradigma, bei dem der Nutzer die Notification-Groups beliebig an die gewünschte Position ziehen kann.

5.2.5.3 Individualisierbarkeit der Notification-Center-Benutzeroberfläche

Neben spezifischen Einstellungen, die den Inhalt des Notification-Centers anpassen, sieht das Konzept auch eine freie Positionierung der Benutzeroberfläche selbst vor. Ohne eine Anpassung des Nutzers, wird das Notification-Center zentriert im oberen Bereich des RCE-Systems dargestellt und ist durch entsprechende

Interaktionsmöglichkeiten, wie der Notification-Bar aus Kapitel 5.3., über die Symbol-Leiste erreichbar. Das Notification-Center besitzt dabei eine fixe Positionierung und wird ohne eine direkte Interaktion nicht dauerhaft angezeigt. Über eine Docking-Funktionalität, kann das Notification-Center jedoch von seiner fixierten Position gelöst, und als externes Fenster dargestellt, sowie per Drag-und Drop beliebig positioniert werden. Diese Art der Darstellung eignet sich insbesondere für Multi-Display-Clients, in denen die Nutzer des Systems einen zusätzlichen Bildschirm einsetzen, um relevante Informationen anzuzeigen.

5.3 Notification-Bar

Auf Grundlage einer Analyse des Awareness-Begriffs in Kapitel 4. und einer ausführlichen Betrachtung gängiger Darstellungsformen für eingehende Ereignis-Benachrichtigungen, konnte anschließend ein Konzept erstellt werden, welches den Nutzern das Eintreten einer Ereignis-Benachrichtigung vermittelt, ohne dessen Aufmerksamkeit unnötig von seiner primären Aufgabe abzulenken. Der Einsatz der so genannten „Notification-Bar“ stellt neben dem Notification-Center einen wesentlichen Bestandteil des konzeptionierten Benachrichtigungs-Systems dar. Zwar erfolgt die Darstellung aller eingehenden Informationen primär über das in Kapitel 5.2. definierte Notification-Center, allerdings muss dieses, sofern es nicht als externes Fenster angepasst wurde, bewusst durch eine Nutzer-Interaktion angezeigt werden. Das Konzept der Notification-Bar sieht hingegen eine Informationsquelle vor, die den Nutzern ein Bewusstsein über eintretende Ereignisse vermitteln soll, ohne dass dieser aktiv in eine Interaktion mit dem System treten muss. Der Nutzer kann sich anhand der Notification-Bar je nach Darstellung darüber informieren, ob er neue Ereignis-Benachrichtigungen erhalten hat oder die zuletzt eingegangene Ereignis-Benachrichtigung lesen, ohne das Notification-Center bewusst aufrufen zu müssen. Der Entwurf der Gestaltungslösungen für diese Darstellungen erfolgte dabei unter Berücksichtigung des Aufmerksamkeit-Nutzen-Modells nach D. Scott McCrickard (siehe Kapitel 4.1.2.). Für die Notification-Bar, definiert dieses Konzept zwei mögliche Darstellungsformen, die sich über die allgemeinen Notification-Einstellungen auch kombinieren lassen sollen:

Darstellung als Message-Bar

Die Darstellung als Message-Bar orientiert sich an der allgemeinen Benachrichtigungs-Syntax, wie sie in Kapitel 5.2.2. beschrieben wurde. Es wird dabei stets nur eine

eingehende Ereignis-Benachrichtigung gleichzeitig angezeigt. Eingeleitet wird eine Benachrichtigung durch eine grafische Repräsentation, die es dem Nutzer ermöglichen soll, eine allgemeine Klassifizierung der eingehenden Benachrichtigung vorzunehmen, ohne diese aktiv lesen zu müssen. Dadurch soll realisiert werden, dass der Nutzer die Aufmerksamkeit nur dann bewusst auf die dargestellte Benachrichtigung lenkt, wenn diese ihm in seinem aktuellen Kontext als relevant erscheint. Die Klassifizierung eingehender Benachrichtigungen orientiert sich dabei an den in Kapitel 5.1.2. identifizierten Notification-Groups.

Ereignis-Benachrichtigungen die eine bestimmte Komponente der RCE-Umgebung betreffen (z.B. die Verfügbarkeit einer verteilten Komponente), werden mit dem entsprechenden Icon der Komponente initialisiert (siehe Abbildung 30). Eine ähnliche Art der Darstellung, wird unter anderem auch in mobilen Betriebssystemen wie Apple's iOS oder dem Android-OS eingesetzt, um Nutzer über eingehende Benachrichtigungen zu informieren. Durch einen Klick auf eine eingegangene Nachricht über die Notification-Bar, gelangt der Nutzer direkt in das Notification-Center, in dem die entsprechende Benachrichtigung dann bereits ausgewählt ist. Der Vorteil einer solchen Darstellung liegt insbesondere darin, dass der Nutzer Informationen wahrnehmen kann, ohne dass diese relevante Elemente der Benutzeroberfläche verdecken und ihn bei der Aufgabenbewältigung beeinträchtigen. Der Nutzer kann so bereits nach Eingang einer Ereignis-Benachrichtigung ein Bewusstsein über relevante Ereignisse entwickeln, ohne in eine direkte Interaktion mit dem Notification-Center treten zu müssen.

Für einen Kontext, in dem viele Ereignis-Benachrichtigungen innerhalb kurzer Zeit aufeinander folgen, eignet sich diese Darstellung jedoch nur begrenzt, da zeitgleich immer nur eine Ereignis-Benachrichtigung dargestellt wird. Dies kann dazu führen, dass eine relevante Information durch den Eingang einer weiteren (irrelevanten) Ereignis-Benachrichtigung überdeckt wird. Diese Art der Darstellung eignet sich daher primär für Nutzer, die für eine effiziente Aufgabenbewältigung stets über den Eintritt von Ereignissen informiert werden möchten und bei denen die Frequenz der eingehenden Ereignis-Benachrichtigungen nicht zu hoch ausfällt.



Abbildung 30: Gestaltungslösung einer Benachrichtigung in der Notification-Bar

Darstellung als Notification-Counter

Die Darstellung als Notification-Counter stellt über entsprechende Icons die, in dem Notification-Center vorhandenen Notification-Groups, dar und informiert den Nutzer durch Ziffer-Indikatoren darüber, ob innerhalb einer bestimmten Notification-Group neue Benachrichtigungen verfügbar sind. Diese Darstellung eignet sich unter anderem für Nutzer, die sich durch eine Vielzahl eingehender Ereignis-Benachrichtigungen über die Notification-Bar nicht ablenken lassen möchten oder denen die Information ausreicht, dass neue Benachrichtigungen vorhanden sind. Mit einem Klick auf ein Icon des Notification-Counters, wird das Notification-Center mit einer maximierten Ansicht der entsprechenden Notification-Group dargestellt, während die andere Notification-Groups minimiert werden (siehe Abb. 31).

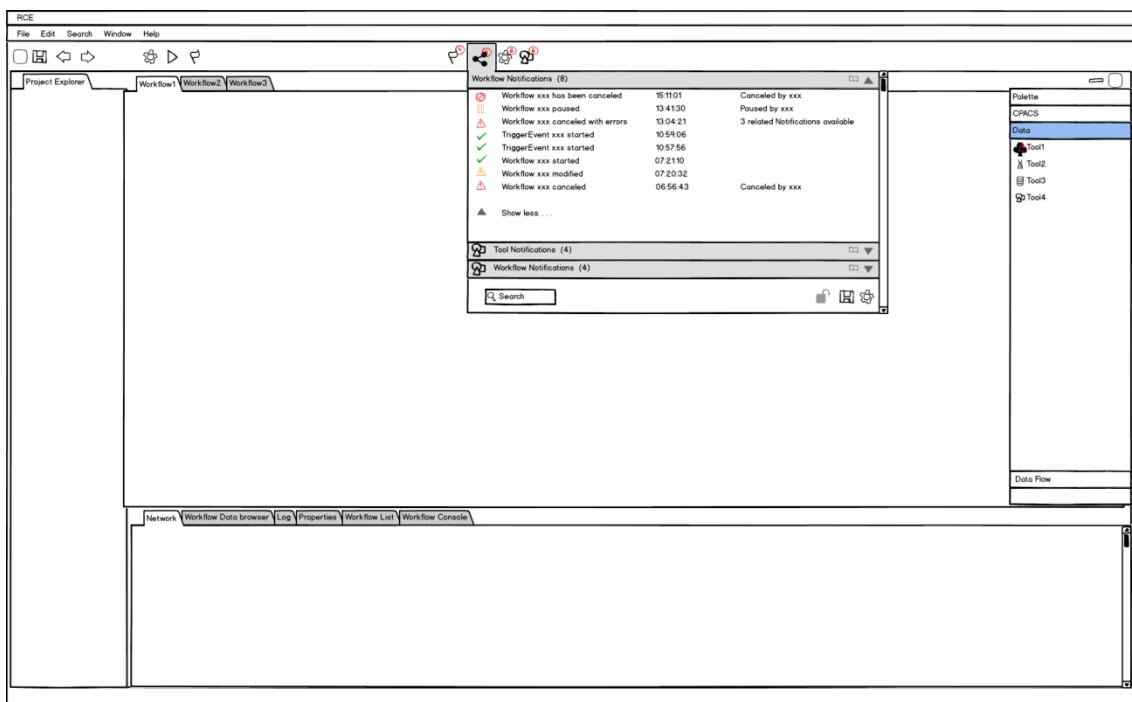


Abbildung 31: Gestaltungslösung der Notification-Counter Darstellung

Die Position der Notification-Bar befindet sich in einem Bereich in der oberen Werkzeugleiste der RCE-Benutzeroberfläche und soll sich dadurch an dem natürlichen Lesefluss eines Menschen orientieren. Welche Art der Darstellung in der Werkzeugleiste angezeigt wird, kann dabei durch den Nutzer individuell angepasst werden. Im Gegensatz zu der Darstellung als Message-Bar, muss der Nutzer hierbei in eine Interaktion mit dem Notification-Center treten, um zu sehen welche neuen Ereignisse eingetreten sind. Er erhält zwar Informationen darüber, in welcher Notification-Group neue Ereignisse eingetreten sind, kann jedoch über diese

Darstellung nicht unmittelbar einschätzen, ob die darin enthaltenen Ereignisse für ihn relevant sind.

5.4 Workflow Trigger-Events

Das Konzept der Workflow Trigger-Events erzielt den Zweck, den Nutzer während einer Workflow-Durchführung physisch von der Benutzeroberfläche des RCE-Systems zu entkoppeln und diesem dennoch ein Monitoring dieser Workflows zu ermöglichen. Die TriggerEvents realisieren dabei individuell vordefinierte Handlungsabläufe des Systems, die auf Basis verschiedener Ereignisse der Simulationsumgebung ausgeführt werden und in entsprechende Benachrichtigungen für den Nutzer resultieren. Die Handlungsabläufe werden dabei durch den Nutzer selbst definiert, wodurch dieser je nach Anforderungen an das Monitoring eigene Benachrichtigungen generieren und über verschiedene Kommunikationskanäle (Notification-Center, E-Mail, SMS, etc.) vermitteln kann. Durch den Einsatz von TriggerEvents soll der Nutzer so ein effizientes Monitoring vornehmen können, während er sich mit anderen Aufgaben befasst, ohne dabei selbstständig regelmäßig den Workflow zur Laufzeit kontrollieren zu müssen. Das Konzept der TriggerEvents besteht dabei im Wesentlichen aus zwei Komponenten:

5.4.1 Trigger

Anhand der Trigger⁵⁰ kann der Nutzer vor oder während der Workflow-Durchführung definieren, unter welchen Bedingungen ein bestimmtes Event⁵¹ ausgelöst wird. Dass das Erzeugen von Triggern auch während der Workflow-Durchführung als sinnvoll erachtet wird, liegt unter anderem daran, dass vor der eigentlichen Durchführung eines Workflows oftmals nicht eindeutig abgewägt werden kann, ob bestimmte Komponenten durch ein Monitoring kontrolliert werden sollen. So kann es vorkommen, dass erst mit der Durchführung einer Simulation anhand eines Workflows unerwartete Werte auftreten, die durch ein angemessenes Monitoring des Nutzers zu beobachten sind. Daher sieht das Konzept der Trigger-Events vor, dass der Nutzer zu jeder Zeit neue Trigger über das System erstellen kann. Zu den im Kontext dieses Projektes als relevant identifizierten Triggern zählen:

⁵⁰ dt.: Auslöser

⁵¹ dt.: Ereignis

Zeitlicher Trigger

Definiert das Auslösen eines Ereignisses zu einem bestimmten Zeitpunkt (Datum und Uhrzeit) oder in bestimmten periodischen Abständen.

Laufzeit-Trigger

Definiert das Auslösen eines Ereignisses, wenn ein Workflow oder ein Tool eine bestimmte Durchlaufzeit überschritten hat.

Workflow- und Tool-Status-Trigger

Definiert das Auslösen eines Ereignisses, wenn ein Workflow oder ein Tool einen bestimmten Status erreicht hat (z.B. running, idle, finished, failed).

Werte-Trigger

Definiert das Auslösen eines Ereignisses, wenn an den Ein- bzw. Ausgängen eines Tools ein bestimmter Wert erreicht oder ein bestimmter Wertebereich überschritten wurde. Hierzu zählt unter anderem auch die Ausgabe von Tools, die über die Konsole ausgegeben werden.

Externer Trigger

Definiert das Auslösen eines Ereignisses, wenn ein verteilter Einsatz von Tools innerhalb der Simulationsumgebung stattfindet. Zu diesen Triggern zählt unter anderem die Durchführung eines Tools auf einer verteilten RCE-Instanz.

5.4.2 Logische Verknüpfung von Triggern

Das Konzept der Trigger-Events ermöglicht, über das Erstellen einzelner Trigger hinaus, zudem eine logische Verknüpfung mehrerer Trigger, wodurch das System spezifischen Anforderungen des Monitorings gerecht werden soll. Im Prozess der Nutzungskontextanalyse und der Entwicklung erster Gestaltungslösungen wurde deutlich, dass das Auslösen von Ereignissen auf Grundlage eines einzelnen Triggers, je nach Kontext kein ganzheitliches und effektives Monitoring gewährleisten kann. So ließen sich Szenarien identifizieren, in denen der Nutzer nur dann durch ein automatisiertes Monitoring Informationen erhalten möchte, wenn mehrere Ereignisse

eingetreten sind. Diese Anforderungen wurden bei der Konzeption der Trigger-Events berücksichtigt und in entsprechende Gestaltungslösungen überführt (siehe Abb. 33). So enthalten die erstellten Prototypen der Trigger-Events-Einstellungen eine „add Trigger“-Funktionalität, die das Hinzufügen weiterer Trigger innerhalb eines Trigger-Events ermöglicht. Möchte ein Nutzer nun mehrere Trigger als Auslöser eines Ereignisses definieren, so kann er zwischen zwei logischen Verknüpfungen (Und bzw. Oder) wählen, anhand derer die Trigger verknüpft werden sollen. Erfolgt eine Kombination zweier Trigger durch den Und-Operator, so wird das entsprechende Ereignis nur dann ausgelöst, sofern beide Trigger erfüllt sind. Durch den Einsatz einer Und-Verknüpfung hat der Nutzer so die Möglichkeit, effektive Bedingungen als Auslöser von Ereignissen zu definieren. Der Oder-Operator hingegen, löst das Ereignis bereits dann aus, wenn nur einer der beiden Trigger erfüllt ist.

Settings

General
RCE
Help
Eclipse
Notifications
- Incoming
- Outgoing
- TriggerEvents

Search

+ add TriggerEvent

- TriggerEvent1

Category: Components

TriggerOn: Increase

Trigger Instance: <local>

Trigger: Tool Status

Logic: AND

TriggerOn: Increase

Trigger: Select

add Trigger

Event: E-Mail

All Instances
Workflow1
Workflow 2

Finished disabled

Please enter Value

Enter Address

Export TriggerEvents

OK

TriggerEvent2 - Trigger on: Tool2 Trigger: Time

Abbildung 32: Gestaltungslösung der Trigger-Event-Einstellungen. Über die "add Trigger" Funktionalität können weitere Trigger durch eine logische Verknüpfung hinzugefügt werden.

5.5 Events

Nachdem der Nutzer einen oder mehrere Trigger gewählt hat, kann er entscheiden, welche Events durch die Trigger ausgelöst werden sollen. Die zur Verfügung stehenden Events sollten dabei kontextbasiert und auf Basis des zuvor gewählten Triggers erfolgen. So ist es unter anderem sinnvoll, dass der Nutzer das Event „Fehlerbericht an Tool-Integrator senden“ nur dann auswählen kann, wenn er zuvor den „Komponentenstatus Trigger“ mit dem Wert „Failed“ festgelegt hat. Anhand der gewählten Trigger soll daher die Wahl der möglichen Events eingeschränkt werden. Durch die kontextabhängige Darstellung möglicher Events soll unter anderem vermieden werden, dass der Nutzer Fehleingaben bei der Erstellung eines Trigger-Events vornehmen kann, was dem Aspekt der Fehlertoleranz nach der DIN EN ISO 9241 Teil 110 entspricht. Neben dem Einsatz der Trigger-Events als Monitoring-Komponente, sieht das Konzept darüber hinaus auch eine rudimentäre Steuerung von Artefakten in dem RCE-System vor. So können zuvor definierte Trigger nicht nur zu einem Benachrichtigungs-Ereignis führen, sondern zusätzlich auch den Status von Artefakten ändern. Insgesamt ergeben sich daraus folgende Events, die durch das Trigger-Event-Konzept abgebildet werden:

Workflow abbrechen

Ein zuvor definierter Trigger führt dazu, dass der entsprechende Workflow abgebrochen wird.

Workflow pausieren

Ein zuvor definierter Trigger führt dazu, dass der entsprechende Workflow pausiert wird

Parameter in Datei schreiben

Ein zuvor definierter Trigger führt dazu, dass die Parameter eines Tools in eine Datei geschrieben werden. Hierzu zählt unter anderem die Konsolen-Ausgabe eines Tools.

E-Mail/SMS senden

Ein zuvor definierter Trigger führt dazu, dass eine Benachrichtigung an die hinterlegte E-Mail-Adresse/Mobilfunknummer gesendet wird. Die technische Realisierbarkeit dieser Ereignisse stellt dabei eine besondere Herausforderung dar, die im Kontext dieser Arbeit nicht beantwortet werden konnte.

Desktop Balloon Message

Ein zuvor definierter Trigger führt dazu, dass der Nutzer eine Benachrichtigung über die System-nativen Balloon Messages erhält.

Popup Message

Ein zuvor definierter Trigger führt dazu, dass der Nutzer eine Benachrichtigung über ein Popup-Fenster erhält.

Notification-Fokus Message

Ein zuvor definierter Trigger führt dazu, dass der Nutzer eine Benachrichtigung in dem Notification-Fokus erhält.

Unter bestimmten Bedingungen kann es zudem erforderlich sein, auf Basis eines Triggers mehrere Events auszuführen. So wurde anderem das Szenario identifiziert, dass ein Nutzer bei Erreichen eines unerwünschten Wertes einer Komponente, sowohl eine Mail an sich senden lässt, als auch zeitgleich den Workflow pausiert um keine unnötigen Rechner-Ressourcen zu verbrauchen. Ähnlich dem Hinzufügen weiterer Trigger, kann der Nutzer daher über die „add Event“-Funktionalität mehrere Ereignisse definieren, die gleichzeitig ausgelöst werden.

Durch die Möglichkeiten, sowohl mehrere Trigger miteinander zu verknüpfen, als auch verschiedene Ereignisse zeitgleich bei Erfüllung von zuvor definierten Triggern auszulösen, soll dem Nutzer auch dann ein effizientes Monitoring gewährleistet werden, wenn dieser nicht aktiv mit dem RCE-System arbeitet. Anhand der Verknüpfung von Triggern durch aussagenlogische Ausdrücke, kann der Nutzer so in einer natürlichen Sprache komplexe Bedingungen für das Ausführen einer oder mehrerer Ereignisse definieren. Mithilfe der Trigger-Events soll der Nutzer zudem bei seiner Aufgabe, ein Monitoring durchzuführen, entlastet werden.

5.5.1 Anlegen eines Trigger-Events

Das Erstellen eines Trigger-Events erfolgt über die Einstellungen der Notifications innerhalb der allgemeinen RCE-Einstellungen. Um den Nutzern eine effiziente Erstellung zu ermöglichen, kann er die Trigger-Events-Einstellungen auch über das Kontextmenü der Artefakte seiner RCE-Instanz aufrufen. In den Einstellungen wählt der Nutzer zunächst das Artefakt aus, welches ein Monitoring erfordert. Da je nach Kontext eines Nutzers die Anzahl der möglichen Artefakte sehr umfangreich ausfallen

kann, lässt sich diese durch eine Vorauswahl der Artefakt-Kategorie gemäß des „Category“-Drop Down Menüs einschränken. Zu den Kategorien zählen neben den Komponenten oder Workflows auch die eigens definierten Tags, die in Abschnitt x.xx. ausführlich beschrieben wurden. Öffnet der Nutzer die Trigger-Event-Einstellungen über den „add Trigger-Event“-Eintrag des Kontext-Menüs eines Artefaktes, so wird in den Einstellungen automatisch ein neues Trigger-Event mit einer Vorauswahl des entsprechenden Artefaktes erstellt. Da ein Artefakt innerhalb der Simulationsumgebung mehrfach instanziiert werden kann⁵², steht dem Nutzer zudem die Option zur Verfügung, die Instanz eines Artefaktes eindeutig zu definieren. Erstellt der Nutzer ein Trigger-Event über das Kontextmenü eines Artefaktes, so wird hierbei die entsprechende Instanz automatisch übernommen. Über den Eintrag „All Instances“ (siehe Abb. 32) kann der Nutzer darüber hinaus alle Instanzen eines Artefaktes überwachen lassen.

Um die verteilten Aspekte des RCE-Systems auch in den Trigger-Events abbilden zu können, kann der Nutzer anschließend die Instanz festlegen, von der aus das erstellte Trigger-Event ausgelöst werden soll. Diese Anforderung konnte unter anderem im Rahmen einer Evaluation (siehe Kapitel 6) erhoben werden. Das definieren einer Trigger-Instanz ist dabei unter anderem notwendig, wenn der Nutzer seinen eigenen Client ausschaltet, aber dennoch über ein Trigger-Event informiert werden will. In diesem Fall kann der Nutzer über die Einstellungen eine andere Instanz außer dem eigenen Client wählen, von der aus das Trigger-Event ausgelöst wird. Wählt der Nutzer keine Instanz selbstständig aus, so wird diese Einstellung automatisch durch das System vorgenommen. In diesem Fall wird die Instanz des entsprechenden Artefaktes als Auslöser des Trigger-Events festgelegt. Legt der Nutzer ein neues Trigger-Event über ein Artefakt in der Durchführung an, so wird als Trigger-Instanz jene Instanz verwendet, auf der das Artefakt durchgeführt wird.

Nach der Auswahl des gewünschten Artefaktes, definiert der Nutzer die eigentliche Bedingung, die dann zu der Ausführung eines Events führt. Der Prototyp sieht dabei insgesamt drei Parameter-Bereiche vor, die eine Bedingung definieren. Während der erste Parameter die Kategorie der Bedingung festlegt⁵³, stellen die anderen Parameter je nach Art einer Bedingung die Eingabe der notwendigen Wertebereiche zur Verfügung. Um Fehleingaben eines Nutzers nach dem Aspekt der Fehlertoleranz⁵⁴ vorzubeugen, hängt die Eingabe möglicher Werte von der zuvor gewählten Ereignis-Kategorie ab. Wählt der Nutzer als Kategorie z.B. eine zeitliche Bedingung, so kann er in den nebenstehenden Parametern nur Datums- sowie Zeitangaben vornehmen.

⁵² Z.B. ein Tool in mehreren Workflows

⁵³ Wie beispielsweise den Tool-Status, die Dauer eines Tools oder den Workflow-Status

⁵⁴ DIN EN ISO 9241 Teil 110

Die erstellten Trigger-Events können durch den Nutzer eindeutig benannt und jederzeit über die Einstellungen bearbeitet sowie gelöscht werden. Durch das Konzept der logischen Verknüpfungen von Triggern, mehreren Events innerhalb eines Trigger-Events und der Vielfalt an möglichen Kombinationen, soll dem Nutzer eine effektive Monitoring-Funktionalität bereitgestellt werden, die sich beliebig der Anforderung des Nutzers anpassen lässt.

5.6 Tool-Verfügbarkeit

Die Entwicklung von Workflows in der verteilten Simulationsumgebung RCE, basiert je nach Kontext einer Aufgabe auf dem Einsatz lokaler sowie verteilt verfügbarer Tools. Ein Tool kann daher insbesondere in einem verteilten Kontext spezifische Zustände einnehmen (z.B. verfügbar, nicht verfügbar, aktualisiert) die sich nachhaltig auf die Aufgabenbewältigung eines Nutzers auswirken. Damit die Nutzer eines vermehrt verteilt realisierten RCE-Systems effizient mit verteilten Komponenten arbeiten können, muss gewährleistet werden, dass diese ein Bewusstsein über diese Zustände der verteilten Komponenten entwickeln können. So kann sich z.B. die Verfügbarkeit verteilter Komponenten nachhaltig auf die Erstellung neuer, sowie die Durchführung bestehender Workflows auswirken. Das Konzept des Benachrichtigungs-Systems sieht daher eine optische Differenzierung verteilt eingesetzter Komponenten vor, anhand derer sich unter anderem deren Verfügbarkeit und verteilter Einsatz identifizieren lässt. Im Prozess der Nutzungskontextanalyse, konnten folgende Zustände erfasst werden:

Komponente verfügbar

Die verteilte Komponente ist verfügbar und kann eingesetzt werden. Anhand dieses Zustandes erkennt ein Nutzer, ob er die Komponente verwenden kann oder ob dadurch seine Workflows zur Laufzeit ggf. nicht erfolgreich terminieren können.

Komponente nicht verfügbar

Eine bereits veröffentlichte Komponente ist nicht mehr verfügbar.

Komponente aktualisiert

Eine Komponente wurde aktualisiert. Dieser Zustand kann unter Umständen zu einer abweichenden Funktionalität der Komponente führen und somit zu Fehlern eines Workflows führen, in dem diese Komponente bereits eingesetzt wird.

Komponente im Einsatz

Die Komponente wird aktuell durch verschiedene Instanzen des RCE-Systems eingesetzt. Diese Information ist insbesondere für Tool-Integratoren sowie Administratoren von Komponenten-Server von Bedeutung, weil sich dadurch unter anderem ein Server-neustart oder Tool-Updates zeitlich besser abstimmen lassen.

5.6.1 Komponenten-Indikatoren

Die verschiedenen Zustände der Verfügbarkeit verteilt eingesetzter Komponenten, werden dem Nutzer durch grafische Indikatoren bereitgestellt. Um angemessene Indikatoren finden zu können, wurden verschiedene Gestaltungslösungen erstellt, die unter anderem der Abbildung 33. zu entnehmen sind.

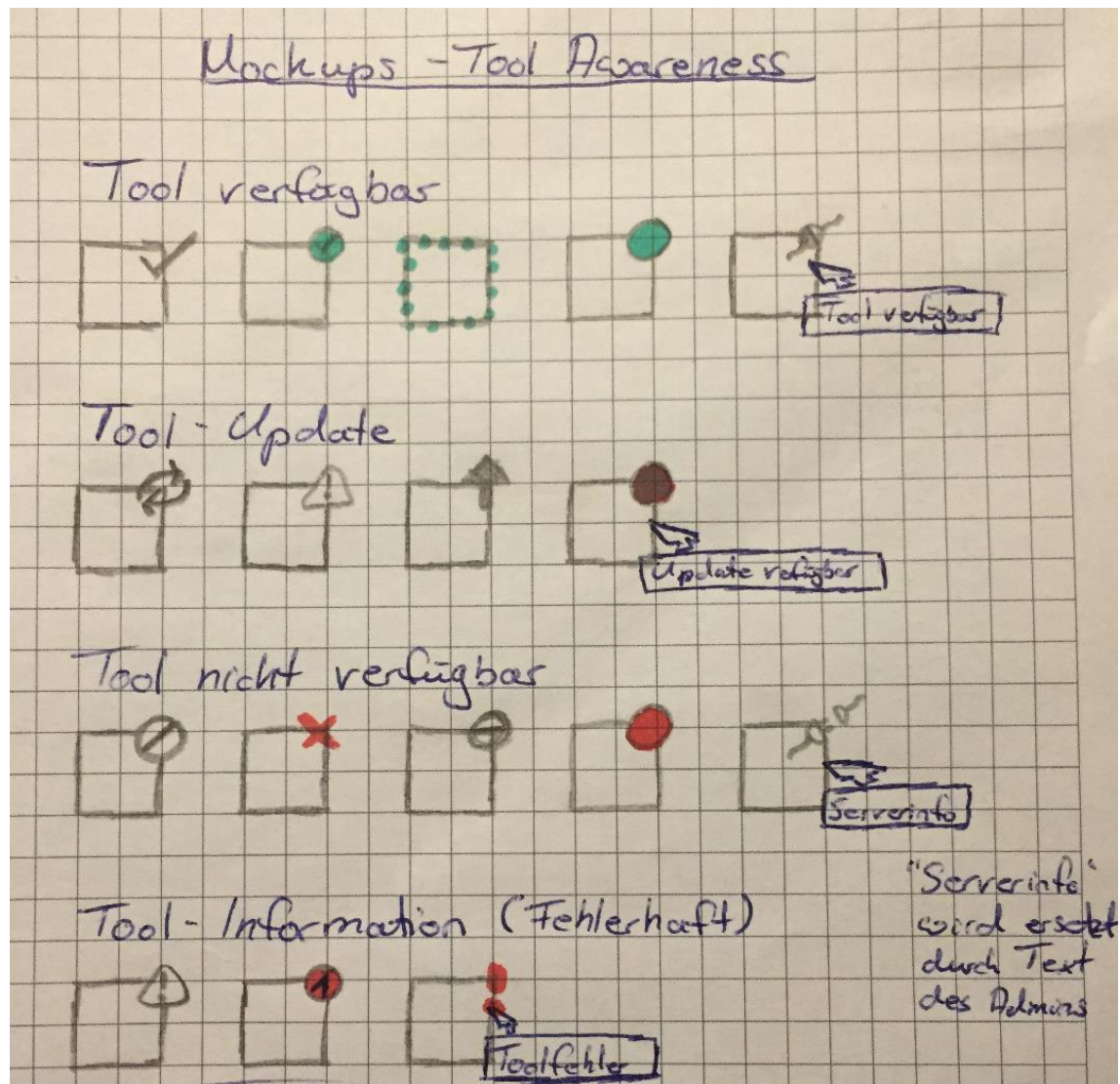


Abbildung 33: Papier-basierte Gestaltungslösung für die Darstellung von Tool-Zuständen

Die Indikatoren bestehen dabei aus dem eigentlichen Icon der jeweiligen Komponente, sowie aus einem Symbol, welches Informationen über die Verfügbarkeit vermitteln soll.

5.6.2 Darstellung von Komponenten-Zuständen

Die Darstellung der Komponenten-Indikatoren erfolgt über zwei verschiedene Informationsquellen:

Tool-Palette

Innerhalb der Tool-Palette einer RCE-Instanz, werden neben lokalen Komponenten, alle verteilt eingesetzten Komponenten aufgeführt. Anhand der Komponenten-Indikatoren

kann der Nutzer schon vor dem Einfügen in den eigenen Workflow erkennen, welche der verteilten Komponenten aktuell verfügbar sind oder welche Komponenten aktualisiert wurden. Bei dieser Darstellung ergibt sich jedoch die Problematik, dass ggf. auch „veraltete“ Komponenten nachhaltig als nicht verfügbar dargestellt werden, obwohl diese nicht mehr erneut zur Verfügung gestellt werden. Über einen längeren Zeitraum, kann sich die Anzahl solcher nicht verfügbarer Komponenten erhöhen und zu einer unübersichtlichen Anzahl in der Tool-Palette führen. Gemäß dieser Annahme ist es sinnvoll, die Komponenten nur für einen beschränkten Zeitraum als nicht verfügbar darzustellen, bevor diese automatisch durch das System aus der Tool-Palette entfernt werden.

Notification-Center

Ändert sich der Verfügbarkeits-Status einer Komponente, so wird dies auch über das Notification-Center mit dem entsprechenden Indikator angezeigt. Der Nutzer kann hierbei zudem weitere Meta-Informationen erhalten, die beispielsweise durch den Tool-Integrator oder den Server-Administrator angelegt wurden (siehe Abb. 20).

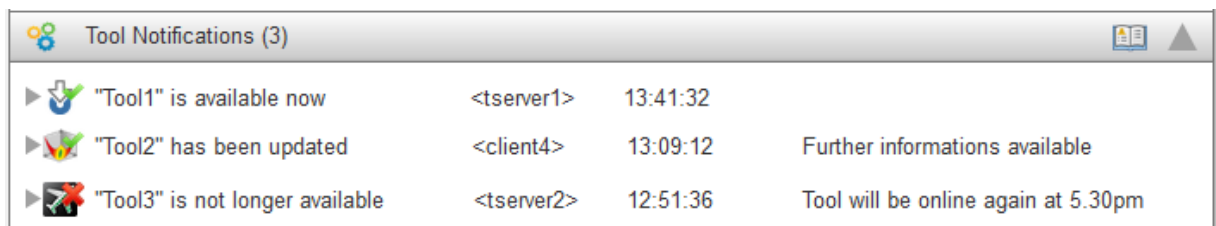


Abbildung 34: Darstellung der Zustände von Tools im Notification-Center

Durch den Einsatz der Komponenten-Indikatoren zur Vermittlung relevanter Zustände verteilter Komponenten, kann der Nutzer seine Aufgaben rechtzeitig anpassen und ggf. laufende Workflows zur Vermeidung eines unerwünschten Abbruchs pausieren bzw. anpassen. Darüber hinaus wird gewährleistet, dass aktiv genutzte veröffentlichte Komponenten, nicht unerwartet entfernt werden, was in den Workflows der jeweiligen Instanzen zu Fehlermeldungen führen kann. So kann allein auf Basis einer grafischen Repräsentation von Komponenten-Zuständen eine effizientere und sicherere Aufgabenbewältigung realisiert werden.

6 Evaluation

Dieses Kapitel beschreibt, wie die Evaluation der in dem Konzept erstellten Prototypen vorbereitet und anschließend durchgeführt wurde. Im Kontext eines Benachrichtigungs-Systems, stellen sich besondere Herausforderungen an eine Evaluation dar, die in diesem Abschnitt näher erläutert werden.

Dieses Kapitel enthält eine Abwägung der Methoden die für die Evaluation berücksichtigt wurden. Die besondere Problematik in der Evaluation dieses Projektes besteht insbesondere darin, dass es sich hierbei primär um eine Evaluation von verschiedenen Benachrichtigungsformen bzw. Kommunikationskanälen handelt. Diese mit Nutzern zu evaluieren erfordert eigentlich funktionale Prototypen in denen die Ereignisse getriggert, und die entsprechenden Benachrichtigungen ausgegeben werden. Da es aufgrund der nur beschränkt zur Verfügung stehenden Zeit und der recht komplexen Programmierung von RCE nicht möglich war, einen funktionalen Prototypen zu entwerfen, sollten daher „Pseudo“-funktionale Prototypen entworfen werden, die sich auch mit den in den Interviews befragten Nutzern evaluieren lassen. Problem dabei ist allerdings, dass sich mit pseudo-funktionalen Prototypen vermutlich keine dynamischen Inhalte „simulieren“ lassen. Das bedeutet, die Evaluation von „eingehenden Nachrichten“ bzw. Benachrichtigungen wird womöglich erschwert⁵⁵.

6.1 Probleme und Risiken

Wie zu Beginn in der Einleitung dieses Kapitels angemerkt, ergeben sich bei der Evaluation von Benachrichtigungs-Systemen besondere Herausforderungen, relevante Aspekte der Gebrauchstauglichkeit zu erfassen. Diesen Herausforderungen liegen bestimmte Probleme und Risiken zugrunde, die in jedem Fall identifiziert werden müssen, um anschließend geeignete Evaluations-Methoden anzuwenden. Der Fokus bei der Vorbereitung einer angemessenen Evaluation liegt dabei insbesondere auf einer Berücksichtigung von Awareness-Aspekten und dem Verhältnis aus Aufmerksamkeits-Kosten und dem Nutzen bei dem Einsatz eines Benachrichtigungs-Systems nach McCrickard. et. al (siehe Kapitel 4). Im Folgenden wird ausführlicher auf relevante Aspekte eingegangen, die bei der Evaluation im Rahmen dieses Projektes und bei der Evaluation von Benachrichtigungs-Systemen allgemein zu berücksichtigen sind.

⁵⁵ Für eine Simulation eines Benachrichtigungs-Systems, sollten die Benachrichtigungen „scheinbar“ zufällig eintreten

7.1.1. Ressourcen bei der Evaluation

Je nach Projekt oder Fokussierung bei einer Evaluation von Gestaltungslösungen, ergeben sich besondere Anforderungen an die benötigten Ressourcen. Als Vorbereitung einer Evaluation ist es daher erforderlich, die zur Verfügung stehenden Ressourcen zu identifizieren. Durch dieses Vorgehen soll gewährleistet werden, dass geeignete Methoden je nach Kontext der Evaluation gewählt und die vorhandenen Ressourcen effizient genutzt werden können. Die Analyse der gegenwärtigen Ressourcen ist daher nicht nur im Rahmen dieses Projektes sinnvoll, sondern sollte generell in Vorbereitung an eine Evaluation jeden beliebigen Projektes vorgenommen werden. So sind insbesondere folgende Ressourcen bei einer Evaluation der in der Konzeptphase erstellten Prototypen zu beachten und wurden im Rahmen dieses Projektes diskutiert.

6.1.1.1 Nutzer als Ressource

Maßgeblich bei einer Evaluation von Prototypen ist insbesondere, in wie fern die tatsächlichen respektive potenziellen Nutzer des späteren Systems aktiv in den Prozess der Evaluation einbezogen werden können. Grundsätzlich ist es ratsam, den Nutzer als Ressource mit in eine Evaluation einzubringen, sofern dies möglich ist. Der Nutzer stellt dabei eine wertvolle Quelle an Informationen dar, indem er während der Evaluation seine Kritik und Vorstellungen äußern und so ein qualitatives Verständnis über den Nutzer mit dessen Wünschen und Anforderungen vermitteln kann. Im Kontext dieses Projektes bestand das wesentliche Problem darin, dass viele der Nutzer die RCE aktiv einsetzen und als potenzielle Benachrichtigungs-System-Anwender in Betracht kommen, an verschiedenen Standorten arbeiten. Daher gestaltete es sich im Anbetracht des nur knappen zeitlichen Rahmens schwierig, alle Nutzer während der Evaluation zu berücksichtigen. Dennoch bestand die Möglichkeit, einen Teil der Nutzer, die auch aktiv am Prozess einer zuvor durchgeführten Nutzungskontextanalyse⁵⁶ teilnahmen, in die eigentliche Evaluation mit einzubeziehen. Der Nutzer als mögliche Ressource kann somit bei der Vorbereitung der eigentlichen Evaluation berücksichtigt und aktiv eingebunden werden.

6.1.1.2 Gestaltungslösungen als Ressource

Eine weitere relevante Ressource stellt die Form der zur Verfügung stehenden Gestaltungslösungen dar. Wie in Kapitel 2.3. angesprochen, existiert eine Vielzahl an möglichen Ausprägungen von Gestaltungslösungen, die von ersten abstrakten

⁵⁶ Quelle Nutzungskontextanalyse

Entwürfen in Form von Scribbles, über detaillierte Mockups bis hin zu Click-Dummies oder funktionalen Prototypen reichen können. Die Art der verfügbaren Gestaltungslösungen kann sich dabei maßgeblich auf die durchzuführende Evaluation auswirken. An dieser Stelle sei allerdings anzumerken, dass die vorhandenen Gestaltungslösungen die zur Auswahl stehenden Evaluations-Methoden nicht zwingend einschränken müssen. Lassen es die zeitlichen Ressourcen zu, so ist es auch denkbar, dass angemessene Prototypen explizit für ein geplantes Vorgehen der Evaluation erstellt werden. Hierbei muss dann zunächst geprüft werden, in wie fern es sinnvoll und zeitlich möglich ist, neue Prototypen gemäß einer Evaluations-Methode zu generieren.

6.1.1.3 Zeit als Ressource

Bevor ein geeignetes Vorgehensmodell für die Evaluation gewählt und entsprechend des Untersuchungsgegenstandes angepasst wird, muss zunächst berücksichtigt werden, wie viel Zeit für die Vorbereitung sowie der eigentliche Durchführung der Evaluation zur Verfügung steht. Mittlerweile existiert eine Vielzahl an Evaluations-Methoden, die sich für verschiedenste Projekte anbieten. Aus diesem Grund, sollte vor einer Festlegung des Ablaufes bei der Evaluation untersucht werden, wie viel Zeit für diese beansprucht werden soll und welche Methoden sich demnach anbieten. Insbesondere empirische Evaluations-Methoden, in denen eine Vielzahl an Nutzern aktiv mit in die Evaluation eingebunden werden können dabei Zeit-intensiv sein, weshalb rechtzeitig eine genaue Abwägung vorzunehmen ist. Vielen der vorgestellten Evaluations-Methoden werden deshalb auch grundlegend Informationen über den zeitlichen Aufwand bei der Durchführung in der Dokumentation beigelegt⁵⁷.

6.1.1.4 Erfahrung als Ressource

Die gesammelten Erfahrung mit den verschiedensten Evaluations-Methoden stellt ebenfalls eine Ressource dar, die nicht außer Acht gelassen werden sollte. Je nach Vorgehen, erfordert die Evaluation eine ausführliche Vorbereitung bzw. Einarbeitung in die einzelnen Prozessstufen, weshalb der zeitliche Aufwand hierbei nicht zu unterschätzen ist. Dies wurde auch im Rahmen dieses Projektes berücksichtigt.

⁵⁷ Siehe z.B. die Lektüre x.xx

7.1.2. Evaluation von Awareness-Aspekten

Wie in Kapitel 4 angedeutet, stellt die Awareness innerhalb einer verteilten Simulationsumgebung oder CSCW-Systemen eine essentielle Komponente dar, die maßgeblich zu einer effizienten Aufgabenbewältigung einer Gruppe beitragen kann. Die in der Konzeptphase erstellten Prototypen berücksichtigen daher auch Mechanismen, die den Nutzern des Systems ein Bewusstsein über den relevanten Gruppenkontext vermitteln sollen. Während die Erstellung von Prototypen, die Awareness-Aspekte berücksichtigen sollen, noch verhältnismäßig einfach fallen mag, stellt eine Evaluation dieser eine besondere Herausforderung dar. So stellt sich unter anderem die Frage, wodurch sich die Awareness eines Nutzers überhaupt auszeichnet bzw. welche Faktoren zu einer „guten“ Awareness beitragen. Ein Ziel der Forschung auf dem Gebiet der CSCW-Systeme befasst sich zudem damit, wie den Nutzern solcher Systeme ein Bewusstsein über verschiedene Ereignisse vermittelt werden kann, ohne diese wesentlich von ihren primären Aufgaben abzulenken. Möchte man solche Aspekte bei einer Evaluation mit echten Nutzern erfassen, so muss eine entsprechende kollaborative Situation mit dem Eintritt von Ereignissen bewusst provoziert oder, falls dies nicht möglich ist, eine solche Situation simuliert werden.

Je nach Art der zur Verfügung stehenden Prototypen (siehe Abschnitt 7.1.1.) ist eine Simulation von eintretenden Ereignissen und den daraus resultierenden Benachrichtigung im Rahmen einer Evaluation allerdings nur bedingt möglich. Existieren für eine solche Evaluation beispielsweise nur Papier-basierte Prototypen oder detailliertere Mockups ohne jegliche Funktionalität, so lassen sich eintretende Ereignisse innerhalb der simulierten verteilten Umgebung nur schwer abbilden und mit den Nutzern evaluieren. Stehen keine funktionalen Prototypen bereit, anhand derer sich Awareness-Aspekte evaluieren lassen, ist daher die Überlegung erforderlich, wie sich der Eintritt eines Ereignisses und die daraus resultierende Notification realitätsgetreu darstellen lassen. Für dieses Projekt wurde daher in Erwägung gezogen, eine Evaluation auf Basis von „Click-Dummies“⁵⁸ durchzuführen.

6.1.2 Evaluation einer Anpassung eines bestehenden Konzeptes

Ein weiteres kritisches Problem, dass speziell im Kontext dieses Projektes auftritt, ist die Evaluation einer Anpassung eines bereits bestehenden Konzeptes. Viele der in der Konzeptphase dieser Ausarbeitung vorgestellten Interaktionsmöglichkeiten, nach dem Eintritt eines Ereignisses, waren bereits vor dieser Konzeption in der verteilten

⁵⁸ Click-Dummies realisieren „pseudo“-funktionale Prototypen um unter anderem Interaktions-Abläufe simulieren zu können

Simulationsumgebung RCE enthalten. Möchte ein Nutzer beispielsweise weitere Informationen über die Fehlermeldung einer Komponente erhalten, so können diese Informationen in dem aktuell bestehenden RCE-System unter anderem der so genannten „Log-View“ entnommen werden. Stellt man den Nutzer im Rahmen einer Evaluation des neuen Konzeptes nun vor die Aufgabe, sich darüber zu informieren warum eine Komponente einen Fehler verursacht hat, so besteht das Risiko, dass der Nutzer seinem „gewohnten“ Vorgehen folgt und das neue Konzept nicht berücksichtigt. In Projekten, in denen bereits bestehende Funktionalitäten neu konzeptioniert werden, ist es daher oftmals nicht vermeidbar, den Fokus bewusst auf neue Interaktionsmöglichkeiten zu lenken. Dieses Vorgehen kann jedoch in Widerspruch zur Erhebung von Awareness-Aspekten stehen, wenn dem Nutzer ein Fokus bei der Evaluation vorgegeben wird und das Bewusstsein des Nutzers bereits zu Beginn auf die neuen Darstellungsformen gelenkt werden.

6.1.3 Erstellung realitätsnaher Tasks

Da der Schwerpunkt dieser Ausarbeitung, gemäß der in der Einleitung erwähnten Forschungsfragen, auf der Konzeption sowie einer Evaluation eines Benachrichtigungs-Systems für eine verteilte Simulationsumgebung liegt, muss das erstellte Konzept auch durch angemessene Methoden evaluiert werden. Eine besondere Herausforderung besteht in einem solchen Projekt darin, geeignete Tasks⁵⁹ zu definieren, anhand derer eine Evaluation durchgeführt werden kann. Die Problemstellung ist dabei, dass ein Nutzer das Lesen von Ereignis-Benachrichtigung selten als Aufgabe definiert oder diese Aktivität als Aufgabe wahrnimmt. Um eine qualitative Evaluation eines Benachrichtigungs-Systems vornehmen zu können, sollten sich die zu durchlaufenden Tasks daher nicht bewusst auf die Benachrichtigungen beziehen. Aufgabenstellungen wie „Lese die eingehenden Benachrichtigungen zu dem Workflow X“ sind für eine qualitative Evaluation also zu vermeiden. Vielmehr gilt es, realitätsnahe Tasks zu definieren, die den tatsächlichen Kontext der potenziellen Nutzer abdecken. Um dennoch Aspekte des Benachrichtigungssystem evaluieren zu können, sind die Tasks dabei so zu wählen, dass die Nutzer⁶⁰ für eine Bewältigung der gestellten Aufgabe, als Teilaufgabe eine Interaktion mit dem Benachrichtigungs-System durchführen müssen.

⁵⁹ dt.: Aufgabe. Im Kontext einer Evaluation von Gestaltungslösungen sind hiermit Aufgaben der Nutzer gemeint

⁶⁰ Ob fiktiv oder reale Nutzer

6.2 Metriken der Evaluation

Als wesentlicher Bestandteil der Vorbereitung einer Evaluation, steht zunächst die Überlegung, welche Metriken überhaupt relevant sind und erfasst werden müssen. Um den, in der Analysephase (siehe Kapitel 4) betrachteten Konzepten gerecht zu werden, sollen für die Evaluation der Prototypen eines Benachrichtigungs-Systems für RCE folgende Metriken berücksichtigt werden:

Learnability

Die Learnability oder auch „Ease of Learning“⁶¹ beschreibt einen Faktor, der definiert wie leicht die Nutzer eine Interaktion mit dem zukünftigen System erlernen können. Im Kontext dieses Projektes ist hier insbesondere von Interesse, wie schnell sich die Nutzer auf der Benutzeroberfläche des Benachrichtigungs-Systems zurecht finden und wieviel Aufwand nötig ist, die Syntax einer Benachrichtigung zu verinnerlichen.

Comprehension (Verständlichkeit) nach McCrickard et.al.

Die Comprehension wurde als ein wesentlicher Faktor in dem Aufmerksamkeits-Kosten und Nutzen-Modell nach 4.1.2.2.. identifiziert und daher auch in der Konzeptionsphase dieses Projektes berücksichtigt. Die Comprehension definiert dabei, ob eingehende Ereignis-Benachrichtigungen für den Nutzer leicht verständlich sind und, gemäß der Expertise des Nutzers, in einer angemessenen Sprache mit entsprechenden Fachbegriffen vermittelt werden.

Reaction (Reaktion) nach McCrickard et.al.

Die Reaction nach McCrickard et.al. definiert einen Faktor, der angibt in wie fern eine wahrgenommene Ereignis-Benachrichtigung zu einer angemessenen Reaktion durch den Nutzer führt. Dieser Faktor beschreibt somit unter anderem, ob die bereitgestellten Ereignis-Benachrichtigungen für den Nutzer relevant waren und ob diese zu einer Anpassung der eigenen Aufgabenbewältigung führen. Je nach Ereignis in der verteilten Simulationsumgebung RCE ist es erforderlich, dass die Nutzer entsprechend reagieren, um eine gegebene Problemstellung weiterhin effektiv bearbeiten zu können, weshalb dieser Faktor auch bei einer Evaluation zu berücksichtigen ist.

⁶¹ Gemäß der DIN EN ISO 9241 Teil 110

Interruption (Unterbrechung) nach McCrickard et.al.

Die Interruption beschreibt, in wie fern eine eingehende Ereignis-Benachrichtigung zu einer Unterbrechung der primären Aufgabe eines Nutzers führt. Dieser Faktor ist insbesondere dann von Bedeutung, wenn sich ein eingetretenes Ereignis auf eine erfolgreiche Aufgabenbewältigung durch den Nutzer auswirken kann. Sofern eine Unterbrechung der aktuellen Aufgabe erforderlich ist, sollte diese von möglichst kurzer Dauer sein, um dem Nutzer dennoch eine effiziente Bewältigung seiner Aufgabe zu ermöglichen.

Awareness

Ein Faktor der sich verhältnismäßig schwer erfassen lässt bezeichnet die Awareness eine Nutzers. Ziel der Awareness im Kontext verteilter Systeme ist es insbesondere, den Nutzern ein Bewusstsein über relevante Ereignisse zu vermitteln, ohne dass diese Nutzer bewusst ihre Aufmerksamkeit für einen längeren Zeitraum auf die entsprechenden Ereignis-Benachrichtigungen lenken muss. Durch die Awareness als zu messender Faktor während der Evaluation kann so erhoben werden, in wie fern ein Nutzer seine Aufmerksamkeit als begrenzte Ressource für die Wahrnehmung einer Ereignis-Benachrichtigung einsetzt.

Jede dieser Metriken stellt dabei besondere Ansprüche an die durchzuführende Evaluation, weshalb sich diese nicht alle gleichermaßen durch nur eine Evaluations-Methode erheben lassen. Im folgenden Abschnitt wird ausführlicher auf die betrachteten Evaluations-Methoden eingegangen und eine Abwägung vorgenommen, in wie weit sich die Metriken in diesen Methoden abbilden lassen. Grundlegend erfolgte dabei eine Betrachtung zweier Ansätze der Evaluations-Methoden

6.3 Empirische Evaluation – Konzept

Bei den empirischen Methoden einer Evaluation im Kontext der Mensch-Computer-Interaktion, werden die potenziellen bzw. zukünftigen Nutzer des Systems aktiv in den Prozess der Evaluation einbezogen. Es erfolgt dabei in der Regel eine Konfrontation der Nutzer mit dem konzeptionierten System, um anschließend Erkenntnisse über kritische Design-Schwächen oder verschiedene Aspekte der Gebrauchstauglichkeit zu erfassen. Zu den empirischen Methoden zählen unter anderem Fragebögen bzw. Interviews, Feldstudien bzw. Beobachtungen oder die so genannten „Card Sortings“. Zu den Anforderungen an eine empirische Evaluation, gemäß der in Kapitel 6.1.1. identifizierten

Ressourcen während eines Projektes, zählt insbesondere die Ressource der Nutzer sowie die Ressource der zur Verfügung stehenden Prototypen. Für eine hinreichend qualitative empirische Evaluation ist es erforderlich, eine angemessene Anzahl an Nutzern für die Evaluation befragen zu können. Da die Nutzer in den empirischen Methoden der Evaluation oftmals mit dem konzeptionierten System konfrontiert werden, ist es zudem ratsam, bereits hinreichend modellierte und (pseudo-)funktionale Prototypen einzusetzen, um unter anderem die Effizienz der Interaktionen zwischen dem Nutzer und dem System zu erfassen.

Zwar lassen sich aufgrund der nur begrenzt zur Verfügung stehenden Zeit und dem Mangel an verfügbaren Nutzern für eine Evaluation im Kontext dieses Projektes keine empirischen Methoden anwenden, dennoch wird in diesem Abschnitt ein Vorgehensmodell beschrieben, dass als Orientierungshilfe und Ausblick weiterführender Arbeiten mit dem Konzept dienen soll. Eine ganzheitliche Evaluation, die alle relevanten Faktoren aus Kapitel 6.2. angemessen erfassen kann, sollte daher nicht allein auf analytischen Evaluations-Methoden beruhen, wie dies in Kapitel 6.4 vorgestellt wird, sondern die in diesem Abschnitt vorgestellte Vorgehensweise ebenfalls berücksichtigen. Bei dieser Beschreibung eines empirischen Vorgehens, erfolgt jedoch keine ganzheitliche Betrachtung aller empirischen Evaluations-Methoden, sondern lediglich eine Beschreibung der als sinnvoll erachteten Methoden. Deren Relevanz für das Projekt wird im Folgenden erläutert.

6.3.1 Beobachtungen

Ein Ansatz der empirischen Evaluation bezeichnet die Beobachtungen der zukünftigen Nutzer während einer Interaktion mit dem konzeptionierten System. Bei diesem Verfahren, erhält der Nutzer verschiedene Aufgabenstellungen, die er unter Verwendung des Systems bewältigen soll. Im Kontext dieses Projektes sieht das vorgeschlagene Vorgehen der empirischen Evaluation den Einsatz einer systematischen Beobachtung⁶² vor, die den Fokus auf das Benachrichtigungs-System legt. Demnach sind auch die entsprechenden Aufgabenstellungen so zu wählen, dass diese einen Einsatz des Benachrichtigungs-Systems erfordern. Das Risiko hierbei besteht insbesondere darin, den Nutzer vor Aufgaben zu stellen, die zwar auf das Konzept des Benachrichtigungs-Systems fokussieren, dabei jedoch nicht dem gewohnten Aufgabengebiet des Nutzers gerecht werden. Die Aufgaben im Rahmen einer empirischen Evaluation sollten dem Nutzer stets als „natürlich“ erscheinen und keinen künstlichen Kontext erzeugen, mit dem der Nutzer sich nicht identifizieren kann.

⁶² Quellenangabe Bort-Döring

Aufgabenstellungen wie „Lege ein Trigger-Event für die Komponente X an“ sollten daher im Rahmen einer effizienten Beobachtung vermieden werden.

6.3.1.1 Think Aloud

Ein mögliches Vorgehen der Beobachtung, welches auch in dem Vorgehensmodell der empirischen Evaluation für dieses Projekt vorgeschlagen wird ist die so genannte „Think aloud“-Methode. Diese ermöglicht es, die kognitiven Vorgänge eines Nutzers bei der Bewältigung einer gegebenen Aufgabenstellung zu interpretieren. Der Nutzer wird bei dieser Methode darum gebeten, seine durchgeführten Handlungsabläufe an dem zu evaluierenden System durch Sprache oder Schrift nach zu erläutern. Ein allgemeines Verständnis der „Think Aloud“-Methode wird im Rahmen dieser Arbeit vorausgesetzt, weshalb diese nicht ausführlicher beschrieben werden soll. Da sich durch dieses Vorgehen ggf. Erkenntnisse über die Handlungsabläufe eines Nutzers gewinnen lassen, soll diese Methode im Rahmen einer Beobachtungen berücksichtigt werden.

6.3.2 Befragungen

Eine der am häufigsten eingesetzten empirischen Evaluationsmethoden zur Datenerhebung⁶³ ist die Befragung der potenziellen Nutzer. Bei einer Befragung kann grundsätzlich zwischen einer mündlichen Befragung in Form eines Interviews (strukturiert bzw. unstrukturiert) und einer schriftlichen Befragung in Form von Fragebögen unterschieden werden. Im Kontext dieses Projektes sieht das Vorgehen bei der Evaluation vor, die Befragungen in Anschluss an eine Beobachtung durchzuführen, weshalb sich der Einsatz einer mündlichen Befragung anbietet. Allgemein sollte vor der Vorbereitung und Durchführung einer mündlichen Befragung, deren wesentliche Ausprägung festzulegen. So lassen sich mündliche Befragungen unter anderem nach Bortz und Döring wie folgt kategorisieren:

- nach dem Ausmaß der Standardisierung (strukturiert – halb strukturiert – unstrukturiert)
- nach dem Autoritätsanspruch des Interviews (weich – neutral – hart)
- nach der Art des Kontaktes (direkt – telefonisch – schriftlich)
- nach der Anzahl der befragten Personen (Einzelinterview – Gruppeninterview – Survey)
- nach der Anzahl der Interviewer (ein Interviewer – Tandem – Hearing)
- nach der Funktion (ermittelnd – vermittelnd)

⁶³ Siehe „Forschungsmethoden und Evaluation“ - Seite

Eine genaue Erläuterung dieser Kategorien soll an dieser Stelle nicht erfolgen, kann jedoch unter anderem der Literatur „Forschungsmethoden und Evaluation“ von Bortz und Döring entnommen werden. Gemäß der zur Verfügung stehenden Ressourcen während dieses Projektes, kann das mündliche Interview nach den soeben vorgestellten Kategorien zugeordnet werden.

So sieht das Vorgehen der empirischen Evaluation für dieses Projekt den Einsatz eines halb strukturierten Interviews vor, in dem der Interview-Leiter zwar gezielte Fragen an die Nutzer des Systems stellt, ohne jedoch vordefinierte Antwortmöglichkeiten bereitzustellen. Durch dieses Vorgehen, kann der befragte Nutzer eigene Antworten formulieren, wodurch sich ggf. kritische Aspekte erfassen lassen, die bei vordefinierten Antwortmöglichkeiten eines strukturierten Interviews nicht adressiert werden. Eine weitere relevante Zuordnung der durchzuführenden mündlichen Befragung ist die Art des Kontaktes während der Befragung. Aufgrund der vielen geografisch verteilten Standorte des DLR, in denen RCE aktiv eingesetzt wird, soll neben dem direkten auch der telefonische Kontakt als mögliche Art für eine Befragung in Betracht gezogen werden. Da es zeitlich ggf. nicht möglich ist, die erforderlichen Beobachtungen sowie mündlichen Befragungen vor Ort an jedem der Standorte durchzuführen, wird so dennoch realisiert, dass mehrere Nutzer-Befragungen durchgeführt werden können. Eine Identifikation einiger geeigneter Fragestellungen für die mündliche Befragung erfolgt in Kapitel x.xx.

6.3.2.1 Interface Claims nach John E. Booker

Ein weiterer Bestandteil des Vorgehens einer empirischen Evaluation orientiert sich an dem Ansatz der “Interface Claims” (Behauptungen über das Interface) nach John E. Booker. Booker schlägt in seiner Ausarbeitung „Usability Testing of Notification Interfaces“ den Einsatz dieser Interface Claims vor, um die Faktoren der comprehension, der interruption und der reaction nach Kapitel 6.2. evaluieren zu können. Die Interface Claims definieren dabei für jede dieser drei Faktoren Behauptungen über das Interface, die die befragten Nutzer entweder bestätigen oder ablehnen sollen. Damit dieses Vorgehen zu einer effizienten Ermittlung dieser Faktoren führen kann, ist es jedoch erforderlich eine ausreichende Anzahl an Nutzern mit diesen Behauptungen zu konfrontieren, um potenzielle Design-Schwächen effektiv zu erheben. Des Weiteren ist es hierbei sinnvoll, auf bereits funktionale Prototypen oder ggf. auch Click-Dummies des zu evaluierenden Systems zurückzugreifen, um so Behauptungen bezüglich einer Interaktion bewerten zu können. Die Durchführung eines Vorgehens nach den Interface Claims, bietet sich insbesondere im Anschluss einer durchgeführten Beobachtung an, in der ein Nutzer bestimmte Aufgabenstellungen mit dem System

bewältigen sollte. Neben den nach Booker zu erhebenden Faktoren, bietet sich der Einsatz der Internet Claims unter anderem auch für die Erhebung von Awareness-Aspekten an. Gemäß diesen Faktoren, können so exemplarisch folgende Behauptungen in einer empirischen Evaluation verifiziert bzw. falsifiziert werden:

Comprehension(Verständlichkeit)

- Ich kann anhand des Notification-Centers nachvollziehen, warum ein Workflow unterbrochen wurde.
- Ich weiß, an welcher Stelle ich Benachrichtigungen über die Netzwerkverbindungen erhalte.
- Ich erkenne, durch welche Instanz ein Ereignis ausgelöst wurde.
- Ich erkenne, wann ein Ereignis ausgelöst wurde.

Reaction

- Ich kann anhand einer dargestellten Fehlermeldung feststellen, was ich zur Behebung des Fehlers tun muss.
- Ich weiß wie ich mir zusätzliche Informationen zu einer Benachrichtigung anzeigen lassen kann.

Interruption

- Ich kann anhand eines Tool-Status identifizieren, wann ich einen Workflow nicht durchführen kann.
- Ich kann anhand einer Tool-Benachrichtigung identifizieren, wann ich einen laufenden Workflow abbrechen muss.

Awareness

- Ich habe wahrgenommen, dass eines meiner verteilt eingesetzten Tools nicht mehr verfügbar ist.
- Ich habe wahrgenommen, dass einer meiner eigenen Workflows, die nicht in der „Runtime-View“ geöffnet waren unerwartet beendet wurde
- Ich habe wahrgenommen, dass eines meiner verteilt eingesetzten Tool aktualisiert wurde

Die soeben erwähnten Behauptungen stellen dabei nur eine Orientierungshilfe dar. Im Rahmen einer empirischen Evaluation des in dieser Arbeit erstellten Konzeptes, sollte daher eine ausführlichere Betrachtung möglicher Interface Claims zur Erhebung der in Kapitel 6.2. identifizierten Faktoren erfolgen. Die zu erstellenden Interface Claims sollten sich dabei an den zuvor durchgeführten Beobachtungen bzw. den darin behandelten Aufgabenstellungen orientieren. So macht es unter anderem wenig Sinn, eine Behauptung nach der Verständlichkeit einer Fehler-Benachrichtigung aufzuführen, wenn in den entsprechenden Aufgabenstellungen keine Fehlermeldung aufgetreten ist.

6.3.2.2 Interface Claims und Aspekte der Awareness

Sollen hingegen Awareness-Aspekte durch Interface-Claims evaluiert werden, bietet es sich an, die Behauptungen nicht gemäß den zuvor behandelten Aufgaben zu formulieren. Daher sieht das Vorgehen der empirischen Evaluation im Kontext dieses Projektes unter anderem vor, bewusst Aufgabenstellungen zu definieren, die den Fokus nicht auf Awareness-Aspekte des Systems legen, während diese anschließend dennoch durch Interface Claims evaluiert werden sollen. So wird der Fokus eines Nutzers während der Bewältigung einer Aufgabe, von den Darstellungen abgelenkt, die Awareness-Aspekte vermitteln sollen (wie beispielsweise der Notification-Bar oder dem Tool Status in der Tool-Palette), um anschließend durch angemessene Interface Claims festzustellen, ob der Nutzer dennoch ein Bewusstsein über diese Aspekte entwickeln konnte. Kann der Nutzer trotz der bewussten Fokussierung auf einen anderen Kontext, die aufgestellten Behauptungen in Bezug auf die Awareness bestätigen, so kann dadurch die Annahme getroffen werden, dass eine effiziente Vermittlung von Awareness-Aspekten stattfindet. An dieser Stelle sei jedoch anzumerken, dass bei einem solchen Vorgehen auch evaluiert werden muss, ob die Darstellung von Awareness-relevanten Informationen über das System nicht zu aufdringlich ist. So besteht unter anderem die Gefahr, dass der Nutzer zwar ein Bewusstsein entwickeln kann, dies jedoch bei einer zu aufdringlichen Darstellung der entsprechenden Informationen, die Aufmerksamkeit eines Nutzers „erzwingt“.

6.4 Analytische Evaluation

Die analytische Evaluation umfasst Methoden, die Im Anschluss an durchgeführte Feldstudien bietet es sich an, weitere relevante Faktoren der Ereignis-Benachrichtigungen im Rahmen eines angemessenen Interviews zu erheben. Der Vorteil dieses Vorgehens ist unter anderem, dass sich die Interviews beliebig skalieren und

somit optimal auf das entsprechende Projekt skalieren lassen. Im Kontext einer Evaluation eines Benachrichtigungs-Systems, muss daher zunächst abgewägt werden, welche Form eines Interviews angemessen ist.

6.4.1 Cognitive Walkthrough

Der Cognitive Walkthrough nach Cathleen Wharton⁶⁴ zählt zu den analytischen Evaluations-Methoden und setzt dabei auf die Erfahrungen von Usability-Experten. Dieser versucht sich im Rahmen einer Evaluation in die Lage potenzieller Nutzer zu versetzen und die Aufgabenbewältigung gemäß den erstellten Prototypen an Gestaltungslösungen zu durchlaufen. Der Cognitive Walkthrough nach Wharton deckt dabei primär den Aspekt des „ease of learning“ als ein Attribut der Gebrauchstauglichkeit ab. Bei einer Durchführung können jedoch beiläufig auch allgemeine Design-Probleme adressiert werden, die nicht im Fokus der Evaluation lagen. Der Cognitive Walkthrough wurde seit seiner Veröffentlichung 1990 mehrmals angepasst, weshalb inzwischen verschiedene Versionen existieren, die auf unterschiedliche Projekt-Ausprägungen und Vorkenntnisse der Durchzuführenden angepasst sind. Das Vorgehen zeichnet sich besonders dadurch aus, dass Nutzer bei einer Evaluation nicht aktiv teilnehmen müssen und eine Evaluation in verschiedenen Konzeptphasen durchgeführt werden kann. So erfordert der Cognitive Walkthrough keinen funktionalen Prototypen für eine Evaluation und kann bereits auf Basis von groben Entwürfen wie Scribbles oder ersten Mockups zu wertvollen Erkenntnissen führen. So können schon in einer frühen Konzeptionsphase wichtige Probleme und erste Design-Schwächen der erstellten Mockus adressiert werden. Als Ausgangsbasis des Cognitive Walkthroughs lassen sich User Profiles oder Personas anwenden, die beispielsweise im Rahmen einer Nutzungskontextanalyse erstellt wurden. Durch eine Betrachtung dieser, soll sich der Durchzuführende besser in die Rolle des potenziellen Nutzers versetzen können.

Um gemäß der Anforderungen an das jeweilige System die erstellten Prototypen evaluieren zu können, ist es als nächster Schritt erforderlich, angemessene Aufgaben, die so genannten „Tasks“ zu definieren, die ein Nutzer durch Unterstützung des Systems bewältigen soll. Für eine qualitative Evaluation durch den Cognitive Walkthrough sollten diese Tasks dabei den tatsächlichen Aufgaben eines potenziellen Nutzers entsprechen und möglichst „natürlich“ wirken. Das bewusste Fokussieren auf die zu evaluierenden Prototypen ist daher zu vermeiden, um die Ergebnisse der Evaluation nicht zu verfälschen. Die erstellten Tasks sind als nächster Schritt in konkrete „Action-Sequences“ zu unterteilen, die ein Nutzer durchlaufen muss, um die

⁶⁴ Wharton C. 1990

gegebene Aufgabenstellung zu bewältigen. Anhand der erstellten Prototypen erfolgen anschließend eine Betrachtung der einzelnen Action-Sequences sowie eine Analyse, ob eine Bewältigung dieser durch die Prototypen hinreichend abgedeckt wird. Um die Evaluation zu erleichtern, bietet es sich an, für jede Action-Sequence angemessene Fragestellungen zu berücksichtigen, die sich auf die potenziellen Nutzer beziehen. Wharton et. al. schlagen dabei die folgenden vier Fragestellungen vor:

„Wird der Nutzer versuchen, den richtigen Effekt zu erzielen?“

„Wird der Nutzer wahrnehmen, dass eine angemessene Aktion zur Verfügung steht?“

„Wird der Nutzer die Interaktionsmöglichkeiten mit dem Effekt, den der Nutzer erzielen möchte in Verbindung bringen können?“

„Wenn die richtige Aktion durchgeführt wurde, wird der Nutzer erkennen, dass er einen Fortschritt bei seiner Aufgabenbewältigung gemacht hat?“⁶⁵

Diese Fragestellungen sind allerdings lediglich als Orientierungshilfe gedacht und können, je nach Projektausprägung, beliebig variieren. An dieser Stelle sei anzumerken, dass der soeben vorgestellte Prozessablauf des Cognitive Walkthrough nur einen allgemeinen Überblick vermitteln soll. Für eine ausführliche Auseinandersetzung mit den einzelnen Prozessstufen empfiehlt es sich, die entsprechende Ausarbeitung von Wharton et. al. zu betrachten.

Im Kontext dieses Projektes, bietet sich der Cognitive Walkthrough als eine Möglichkeit an, um die erstellten Prototypen aus der Konzeptphase zu evaluieren. Insbesondere die als kritisch erachteten Ressourcen der nur begrenzt zur Verfügung stehenden Zeit sowie mangelnder Nutzer, die für eine Evaluation befragt werden können, sprechen für den Einsatz eines Cognitive Walkthroughs. Der Vorteil dieser Methode besteht zudem darin, dass anhand zuvor erstellter User Profiles bzw. Personas verschiedene Nutzer-Ausprägungen bei einer Evaluation berücksichtigt werden können, die durch „echte“ Nutzer ggf. nicht abgedeckt werden. So wurde unter anderem durch die zuvor durchgeführte Nutzungskontextanalyse deutlich, dass die Nutzer die potenziell für eine Evaluation befragt werden könnten, überwiegend ausgeprägte Erfahrungen über das RCE-System besitzen. Eine Evaluation von Gestaltungslösungen allein auf Basis erfahrener Nutzer durchzuführen, sollte vermieden werden und birgt das Risiko, wesentliche Aspekte der Gebrauchstauglichkeit⁶⁶ zu vernachlässigen.

⁶⁵ Wharton C. 1990, S.3

⁶⁶ Wie „ease of learning“, „predictability“, etc.

6.4.2 Groupware Walkthrough

Der „Groupware Walkthrough“ nach David Panelle und Carl Gutwin⁶⁷ erweitert den Cognitive Walkthrough um kollaborative Aspekte, um verteilte Systeme oder CSCW-Systeme allgemein effizienter evaluieren zu können. Das Vorgehen sieht dabei in Analogie an die „Tasks“ des Cognitive Walkthrough die Erstellung eines so genannten „Task-Models“ vor, in dem die Zusammenhänge aus lokalen und verteilten Tasks verdeutlicht werden. Um identifizieren zu können, welche Tasks kollaborative Aktivitäten enthalten, erfolgt zunächst eine Identifikation von Szenarien, die die individuelle Sicht einzelner Nutzer einer Gruppe beschreiben. Innerhalb eines Szenarios bewältigt der jeweilige Nutzer unter Verwendung des Systems dabei eine bestimmte Aufgabe, die sowohl lokale als auch kollaborative Interaktionen beinhalten kann.

Innerhalb eines erstellten Szenarios kann darauf hin identifiziert werden, wann ein Nutzer von einer lokalen auf eine kollaborative Interaktion wechseln muss. Die kollaborative Interaktion innerhalb eines Szenarios lässt sich anschließend mit dem entsprechenden Szenario des anderen Nutzers, mit dem eine Interaktion eingegangen wird, verknüpfen. Sofern eine solche kollaborative Verknüpfung zweier Szenarien möglich (und sinnvoll) ist, bieten sich diese für eine Durchführung des Groupware Walkthroughs an. In einem Task Model findet daraufhin eine Definition relevanter Sub-Tasks (Teil-Aktivitäten) statt, die sich aus solchen Szenarien herleiten lassen. Diese Sub-Tasks können in individuelle bzw. lokale Sub-Tasks oder in kollaborative Sub-Tasks eingeteilt werden. Die Granularität der einzelnen Sub-Tasks kann dabei je nach Anforderungen an die Genauigkeit der Evaluation variieren, indem ein Sub-Task wiederum in weitere Sub-Tasks unterteilt wird.

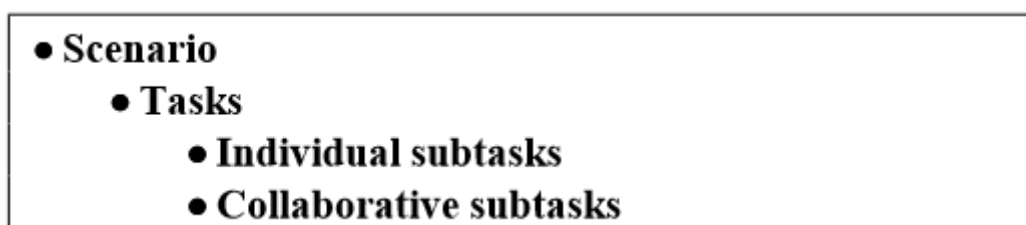


Abbildung 35: Unterteilung der Tasks eines Szenarios nach dem Groupware Walkthrough

⁶⁷ Panelle D. 2002

Kollaborative Sub-Tasks werden innerhalb des Task Models durch eine Nutzer-übergreifende Verbindung verdeutlicht, während sich lokale Tasks stets auf einen individuellen Nutzer beziehen und keine „übergreifenden“ Verbindungen zu Tasks anderer Nutzer enthalten. Dieses Vorgehen lässt sich exemplarisch der Abbildung 36 entnehmen, in der ein kollaboratives Szenario zweier Nutzer durch ein Task-Model abgebildet wurde. Ist die Task-Analyse abgeschlossen und das Task Model hinreichend definiert, kann der Groupware Walkthrough unter Verwendung des Task Models und der erstellten Prototypen ähnlich dem Vorgehen des Cognitive Walkthroughs durchlaufen werden. Auch in diesem Vorgehen sind dabei angemessene Fragestellungen zu berücksichtigen, um Aspekte der Gebrauchstauglichkeit evaluieren zu können.

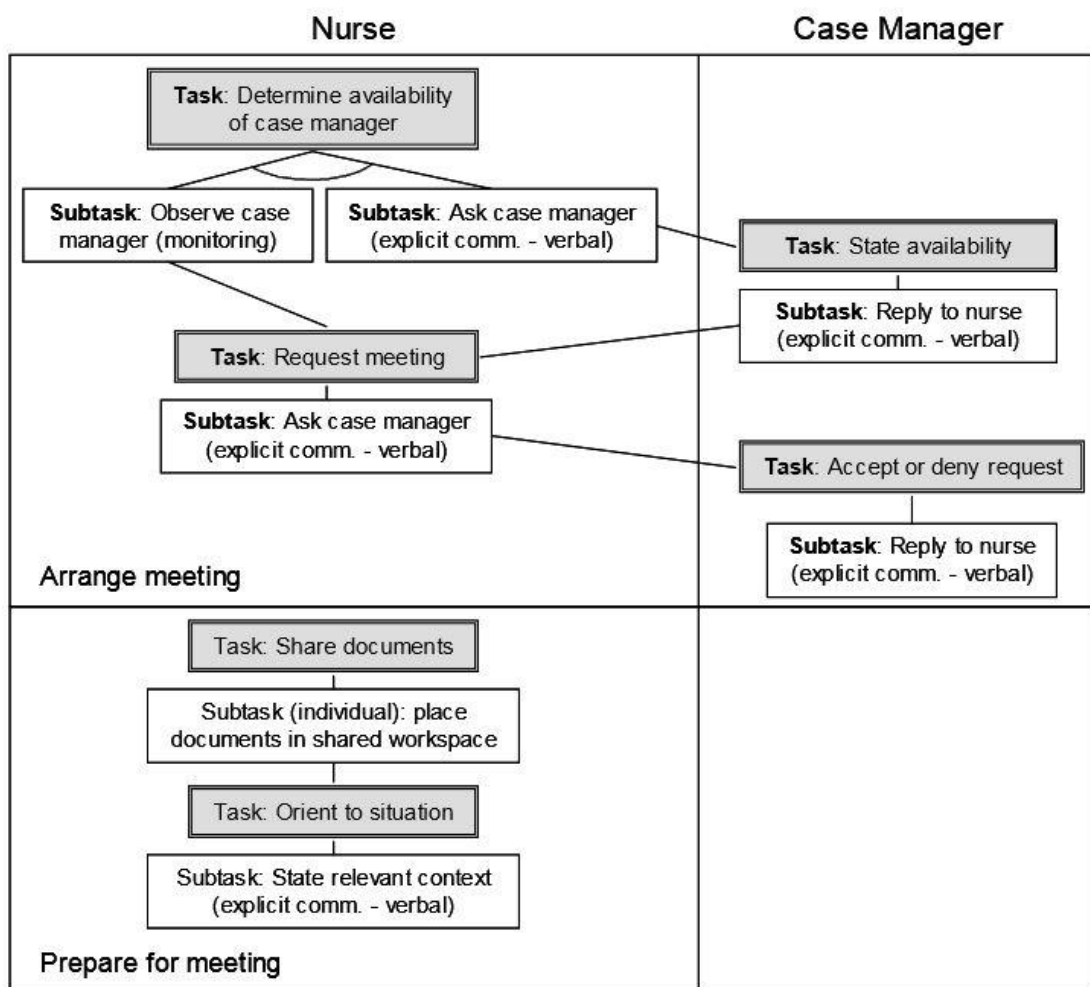


Abbildung 36: Exemplarisches Szenario eines kollaborativen Tasks nach dem Groupware Walkthrough⁶⁸

⁶⁸ Panelle D. 2002, S. 458

Durch den Einsatz des Groupware Walkthroughs kann gewährleistet werden, dass sich auch kollaborative Interaktionen zwischen Nutzern eines CSCW-Systems effizient evaluieren lassen. Für die Evaluation des in dieser Ausarbeitung vorgestellten Konzeptes bietet sich diese Methode allerdings nur bedingt an, da die Erstellung der verschiedenen Szenarien mit einem erheblichen zeitlichen Aufwand verbunden ist. Zudem liegt der Fokus dieser Ausarbeitung darin, die Ereignisse innerhalb einer verteilten Simulationsumgebung anhand des RCE-Systems unter Berücksichtigung gebrauchstauglicher Aspekte zu vermitteln. Eine direkte Interaktion oder die „klassische“ kollaborative Erarbeitung von Artefakten, wie dies nach dem Verständnis des CSCW-Begriffs vorgesehen ist, findet dabei nur bedingt statt. Dennoch können Ereignisse in RCE mitunter dazu führen, dass sich Nutzer direkt austauschen müssen um beispielsweise gemeinsam die Lösung zu einer gegebenen Problemstellung zu erarbeiten. Deshalb ist ein Vorgehen nach dem Groupware Walkthrough nicht grundsätzlich auszuschließen und kann sich für einzelne Tasks durchaus anbieten.

6.5 Pluralistic Walkthrough

Der Pluralistic Walkthrough nach Randolph Bias⁶⁹ beschreibt eine Evaluations-Methode, bei der ein Vorgehen ähnlich des Cognitive Walkthroughs, mit Unterstützung echter Nutzer und Entwicklern des zu betrachtenden Systems durchlaufen wird. Das Vorgehen lässt sich daher nur schwer eindeutig in die Kategorien der analytischen bzw. empirischen Evaluations-Methoden unterteilen, da sowohl Ansätze aus analytischen als auch empirischen Vorgehensmodellen enthalten sind. Der Pluralistic Walkthrough bietet sich unter anderem für Projekte an, in denen rein empirische Evaluations-Methoden aufgrund mangelnder Ressourcen wie Nutzer-Verfügbarkeit oder nicht ausreichender Zeit nur schwer zu realisieren sind, auf die Teilnahme echter Nutzer aber nicht vollständig verzichtet werden möchte. Grundlegend sieht der Pluralistic Walkthrough drei Instanzen vor, die aktiv an einer Evaluation teilnehmen:

- Entwickler des Systems
- Usability-Experten
- Potenzielle Nutzer des Systems

⁶⁹ Bias R. 1994

Die jeweiligen Instanzen können durch eine nicht fest definierte Anzahl an Teilnehmern repräsentiert werden. So ist es unter anderem ratsam, mehrere echte Nutzer des Systems in die Evaluation einzubeziehen, sofern dies möglich ist. Der Pluralistic Walkthrough lässt durch dieses Vorgehen gut auf verschiedene Projekte und die zur Verfügung stehenden Ressourcen skalieren.

Für eine Durchführung des Pluralistic Walkthroughs nimmt jede dieser drei Instanzen zunächst die „Rolle“ eines potenziellen Nutzers ein. Diese Nutzer-Rollen versetzen sich dabei in vordefinierte Szenarien, in denen es jeweils eine bestimmte Aufgabe unter Verwendung des zu evaluierenden Systems zu bewältigen gilt. Das System wird dabei durch verschiedene (nicht funktionale) Prototypen repräsentiert. Da die Methode keinen Einsatz funktionaler Prototypen vorsieht, bietet es sich an, die verschiedenen System-Zustände die bei einer Aufgabenbewältigung eintreten, durch entsprechende Mockups oder „Screenshots“ abzubilden.

Die Teilnehmer notieren anhand dieser System-Zustände jeder für sich, wie sie eine gegebene Aufgabe mit dem System bearbeiten würden. Anschließend erfolgt eine Betrachtung der durch die Teilnehmer angefertigten Notizen. Dabei beginnt zunächst der Administrator bzw. Entwickler, seine Teil-Aktivitäten zur Bewältigung der gestellten Aufgabe zu erläutern. Während die jeweiligen Instanzen ihre Handlungsabläufe beschreiben, sollen Diskussionen vermieden werden. Nach dem Entwickler, folgen eine Beschreibung des Vorgehens der tatsächlichen Nutzer und anschließend, der eigentlich für die Prototypen vorgesehene Ablauf des Usability-Experten. Anhand dieses Vorgehens, lassen sich drei verschiedene Perspektiven der Aufgabenbewältigung anhand der Prototypen erfassen. So lässt sich effizient ermitteln, in wie weit die einzelnen Vorstellungen einer Aufgabenbewältigung anhand der Prototypen variieren können und wo ggf. eine Anpassung dieser notwendig ist. Für den Entwickler des Systems bietet der Pluralistic Walkthrough zudem den Vorteil, dass dieser ein besseres Verständnis über die Nutzer und deren Wünsche erhalten kann. Erst nachdem jede der drei Instanzen ihren Handlungsablauf beschrieben haben. Folgt eine Diskussion über die abweichenden Ansätze.

Mit dieser Methode lassen sich jedoch nur bedingt Erkenntnisse über die Effizienz des jeweiligen Systems, die allgemeine Gebrauchstauglichkeit und die Navigation über das gesamte Interface gewinnen. Ähnlich dem Cognitive Walkthrough, fokussiert auch dieses Vorgehen primär auf den Aspekt des „ease of learning“, wobei in einer anschließenden Diskussion aller Teilnehmer durchaus auch allgemeine Probleme hinsichtlich der Gebrauchstauglichkeit identifiziert und behandelt werden können. Um die Qualität dieser Evaluations-Methode zu erhöhen und auch bewusst andere Aspekte der Gebrauchstauglichkeit abzudecken, bietet es sich an, die abschließende Diskussion durch geeignete Fragestellungen für die echten Nutzer vorzubereiten.

Neben dem Cognitive Walkthrough, soll im Kontext dieses Projektes auch eine Evaluation nach dem pluralistic Walkthrough durchgeführt werden. Die Methode eignet sich insbesondere für Szenarien, in denen noch kein hinreichendes Verständnis über den Nutzer mit dessen Kontext und Aufgaben vorhanden ist. So können auch in diesem Projekt Aufgabenstellungen existieren, von denen der Projekt-durchzuführende allein auf Basis narrativen Beschreibungen wie Personas oder Szenarien, keine konkrete Vorstellung einzelner Handlungsabläufe eines Nutzers entwickeln kann. Eine Evaluation nach dem Cognitive Walkthrough würde in einem solchen Szenario ggf. zu verfälschten Ergebnissen führen, was sich nachhaltig auf die Gebrauchstauglichkeit des Systems auswirken kann. Für dieses Projekt bzw. die Evaluation, war die Ressource des Nutzers zwar äußerst begrenzt, dennoch bestand die Möglichkeit, den Pluralistic Walkthrough mit einem echten Nutzer durchzuführen.

7 Planung der Evaluation

Dieses Kapitel befasst sich mit der Planung der Evaluation gemäß der in Kapitel 6 durchgeführten Abwägungen und der eigentlichen Durchführung. Die Planung setzt sich grundlegend aus einer Erstellung angemessener Tasks für die Nutzer sowie aus verschiedenen Fragestellungen zusammen. Die in diesem Kapitel erstellten Tasks und Fragestellungen können dabei sowohl für die analytische, als teilweise auch für die empirische Evaluation zum Einsatz kommen.

7.1.1 Definition der Tasks

In Vorbereitung an eine Evaluation, gilt es zunächst geeignete Aufgabenstellungen zu finden, anhand derer sich die erstellten Prototypen aus der Konzeptphase evaluieren lassen. Wie in Kapitel 6.3. erläutert, sollen die Tasks dabei möglichst realitätsnahe wirken, aber dennoch Aspekte des Benachrichtigungs-Systems beinhalten. Nach einer ausführlichen Betrachtung des RCE-Systems, der identifizierten Aufgaben aus der zuvor durchgeführten Nutzungskontextanalyse sowie der Prototypen aus der Konzeptphase, wurden daher folgende Tasks definiert, die sowohl bei dem Cognitive- als auch bei dem Pluralistic Walkthrough durchlaufen werden sollten:

- Interpretieren, warum ein Workflow unerwartet beendet wurde.
- Ergebnisse eines Workflows „remote“ anzeigen lassen
- Informationen über einen Workflow in den eigenen Fokus rücken um Informationen zu erhalten
- Informationen darstellen lassen, wer das eigens entwickelte Tool einsetzt
- Informationen darstellen lassen, wenn verteilte Tools des eigenen Workflows nicht verfügbar sind
- Für eine eigens definierte Gruppe an Tools, Informationen anzeigen lassen
- Identifizieren, warum ein Workflow nicht mehr fehlerfrei durchläuft

Diese Tasks fokussieren dabei auf Wesentliche Bestandteile des erstellten Konzeptes und sollen keine ganzheitliche Modellierung aller möglichen Aufgaben eines Nutzer beschreiben, in denen das Benachrichtigungs-System eingesetzt wird. So sind unter anderem weitere Aufgaben bzw. Szenarien denkbar, in denen die Nutzer in eine Interaktion mit dem System treten, die allerdings im Rahmen dieser Ausarbeitung aus Gründen der nur begrenzt zur Verfügung stehenden Ressourcen nicht berücksichtigt

wurden. Zu jeden dieser Tasks wurde nun ein entsprechendes Szenario erstellt, sowie Aktions-Sequenzen zur Bewältigung dieser Tasks definiert. Die Aktions-Sequenzen sollen dabei den von dem Nutzer erwarteten Handlungsablauf definieren. Weichen die von dem Nutzer tatsächlich durchgeführten Aktions-Sequenzen von den Erwartungen ab, so ist ggf. eine Anpassung der Interaktionsgestaltung notwendig. Der konkrete Aufbau einer Task-Beschreibung wird wie folgt anhand des Tasks „Interpretieren, warum ein Workflow unerwartet beendet wurde.“ exemplarisch verdeutlicht:

Task: Interpretieren, warum ein Workflow unerwartet beendet wurde.

Scope: Notification-Center, Comprehension

Szenario: Ein Workflow in der Durchführung bricht plötzlich unerwartet ab. Der Nutzer möchte nun herausfinden, weshalb dies geschehen ist.

Action-Sequences:

1. Meldung in der Notification-Bar anklicken
 - 1.1. Pfeil-Icon der Notification-Bar anklicken
2. Notification-Center öffnet sich
3. Kategorie der Workflow-Notifications betrachten
 - 3.1. Suchbegriff (Workflowname) in Filtermaske eingeben
 - 3.2. Suche durchführen
4. Identifizieren, ob weitere Informationen der Benachrichtigung verfügbar sind
5. Weitere Informationen der Benachrichtigung durch Klick auf Pfeil-Icon anzeigen
6. Zugehörige Benachrichtigungen öffnen sich
7. Ursprung des Abbruchs ermitteln

Eine ausführliche Beschreibung der als relevant erachteten Tasks lässt sich dem Anhang dieser Ausarbeitung entnehmen. Der darin enthaltene „Scope“ definiert für jede der Task-Beschreibung, welcher Teil des Konzeptes durch diese Aufgabe abgedeckt werden soll. Diese Definition wurde als sinnvoll erachtet, da einige der erstellten Task-Definition nicht bewusst den Einsatz des Benachrichtigungs-Systems ansprechen, diese jedoch eine Interaktion mit dem System provozieren sollen.

7.1.2 Definieren geeigneter Fragestellungen

Die Identifikation geeigneter Fragestellungen basiert primär auf den in Kapitel x.xx. als relevant erachteten Faktoren. Zwar bestand im Rahmen dieses Projektes nicht die Möglichkeit, eine umfangreiche empirische Evaluation mit einer angemessenen Anzahl potenzieller Nutzer des Systems durchzuführen, jedoch war es möglich das Vorgehen des Pluralistic Walkthroughs exemplarisch mit einem erfahrenen Nutzer des RCE-Systems zu durchlaufen. Im Anschluss an die Durchführung des Pluralistic Walkthroughs, sollen die in diesem Prozess identifizierten Fragestellungen in eine mündliche Befragung einfließen, um neben dem Aspekt des „ease of learning“ auch die Faktoren der comprehension, der reaction, der interruption sowie der Awareness zu evaluieren. Darüber hinaus können die identifizierten Fragestellungen auch bei dem vorgeschlagenen Vorgehen der empirischen Evaluation eingesetzt werden, um eine ganzheitliche Evaluation zu gewährleisten.

Die soeben erwähnten Faktoren, wurden bei der Identifikation angemessener Fragestellungen als taktische Zielsetzungen einer Zielhierarchie der mündlichen Befragung übernommen und bilden gemeinsam das strategische Ziel, die Gebrauchstauglichkeit des konzeptionierten Benachrichtigungs-Systems zu erfassen:

Strategisches Ziel:

Gebrauchstauglichkeit des Benachrichtigungs-Systems erfassen

Taktische Ziele:

- Aspekt der Verständlichkeit (comprehension) erfassen
- Aspekt der Reaktion (Reaction) erfassen
- Aspekt der Unterbrechung (Interruption) erfassen
- Aspekt des Bewusstseins (Awareness) erfassen

Operative Ziele:

- Aspekt der Verständlichkeit erfassen:
 - Verständlichkeit der Benachrichtigungs-Syntax
 - Verständlichkeit der abgebildeten Tool-Zustände
 - Verständlichkeit der Notification-Center-Einstellungen
 - Verständlichkeit der Trigger-Event-Einstellungen
 - Verständlichkeit des Tagging-Systems

- Aspekte der Reaktion erfassen:

- Benachrichtigung vermittelt Notwendigkeit einer Reaktion
- Benachrichtigung vermittelt Möglichkeiten der Reaktion
- Nutzer erkennt Notwendigkeit einer Reaktion
- Aspekte der Unterbrechung erfassen:
 - Benachrichtigung führt zu Änderung in der Aufgabenbewältigung
 - Benachrichtigung erfordert Unterbrechung eines Workflows
 - Benachrichtigung erfordert Pausieren eines Workflows
- Ereignis über potenzielle Abbrüche

- Aspekt des Bewusstseins

- Bewusstsein über Workflow-Aktivitäten
- Bewusstsein über Tool-Verfügbarkeit
- Bewusstsein über eingehende Ereignisse

Aus den operativen Zielsetzungen ließen sich anschließend angemessene Fragestellungen generieren, die je nach Fokus der mündlichen Befragung eingesetzt werden können. So führte z.B. das operative Ziel „Verständlichkeit der abgebildeten Tool-Zustände“ unter anderem zu Fragestellungen wie:

- Woran können Sie erkennen, welchen Zustand ein verteilt eingesetztes Tool aktuell hat?
- Woran erkennen Sie, ob ein von ihnen eingesetztes verteiltes Tool aktualisiert wurde?
- Sind die unterschiedlichen Darstellungen zur Unterscheidung der Tool-Zustände verständlich?

Bei der Erstellung der Fragestellungen wurde darauf geachtet, einen suggestiven Stil zu vermeiden, um die Ergebnisse einer mündlichen Befragung nicht negativ zu beeinträchtigen. Da das Vorgehen der mündlichen Befragung ein halb-strukturiertes Interview vorsieht, wurden zu den identifizierten Fragestellungen keine Antwortmöglichkeiten generiert, so dass der Nutzer eigene Antworten auf die Fragestellungen geben kann. Die Durchführung des Interviews sieht dabei keine vordefinierte Reihenfolge vor, in denen die identifizierten Fragen gestellt werden, sondern kann je nach durchgeführter Aufgabe oder Fokussierung einer Sitzung erfolgen. Auch ähneln sich einige der Fragestellungen inhaltlich und müssen daher nicht alle in einer mündlichen Befragung gleichermaßen beachtet werden. Dies ist nur dann notwendig, sofern die Antwort eines befragten Nutzers als unsicher eingestuft wird und

diese, im weiteren Verlauf des Interviews, durch eine Frage mit ähnlichem Fokus validiert werden soll.

7.2 Bewertung und iterative Überarbeitung

Auf Grundlage einer als angemessen erachteten Vorbereitung der analytischen Evaluation, konnte diese anschließend durchgeführt werden. Mit der Durchführung der Pluralistic- sowie des Cognitive-Walkthroughs, ließen sich mehrere Design-Schwächen des Konzeptes identifizieren und iterativ überarbeiten. Neben grundlegenden Problemen der Gebrauchstauglichkeit, die aufgrund eines teils mangelnden Verständnisses des Nutzungskontextes begünstigt wurden, konnten darüber hinaus auch weniger kritische Aspekte, wie beispielsweise unangemessene Begrifflichkeiten oder falsch interpretierte Symboliken, identifiziert und überarbeitet werden. Die in diesem Kapitel aufgeführten Mängel der Gestaltungslösungen, basieren dabei nicht ausschließlich auf den gewonnenen Erkenntnissen der abschließenden Evaluation, sondern ließen sich auch im Rahmen von Absprachen mit dem RCE-Entwicklerteam identifizieren.

7.2.1 Iterative Anpassung des Notification-Centers

Bei der Durchführung des Pluralistic-Walkthroughs, konnten neben kleineren Design-Schwächen auch neue Anforderungen an das Notification-Center identifiziert werden. Während der Bewältigung eines Tasks wurde so unter anderem deutlich, dass in dem Notification-Fokus auch allgemeine Fehlermeldungen jederzeit dargestellt werden sollten. Zum Zeitpunkt der Durchführung des Pluralistic Walkthroughs, sah das Konzept als mögliche Informationen für den Notification-Fokus lediglich die „Fokus-Tags“, die aktiven Workflows, sowie die Nutzer-Benachrichtigungen vor. Aufgrund der identifizierten Anforderung, auch Fehlermeldungen⁷⁰ über den Notification-Fokus darzustellen, erfolgte eine Anpassung des Konzeptes um die „Error-Messages“. Das Konzept sieht zudem eine optische Differenzierung zwischen Fehlermeldungen und anderen Benachrichtigungen vor, indem Fehlermeldungen vor einem roten Hintergrund dargestellt werden.

Ein weiterer kritischer Aspekt des Notification-Centers, der zu dem Konzept der Notification-Cluster geführt hat, wurde während einer Absprache über erstellte Gestaltungslösungen mit Mitgliedern des RCE-Entwicklerteams erhoben. So schien die grundlegende Darstellung von Ereignis-Benachrichtigungen in dem Notification-Center trotz der Unterteilung in angemessene Kategorien kritisch, wenn eine Vielzahl an

⁷⁰ Z.B. über Workflows oder Tools

Ereignissen innerhalb kurzer Zeit eintreten. Gemäß diesen Szenarios und anhand der identifizierten Anforderung, Ereignisse für den Nutzer nachvollziehbar zu gestalten, sollte das Konzept eingehende Ereignisse, sofern möglich, angemessen zusammenfassen. Die daraus resultierenden Notification-Cluster wurden anschließend sowohl durch den Pluralistic Walkthrough als auch durch den Cognitive Walkthrough evaluiert und als angemessene Gestaltungslösung identifiziert.

Die Durchführung des Cognitive Walkthroughs führte zudem zu einer Anpassung der vorgestellten Benachrichtigungs-Syntax. Diese bestand zunächst nur aus dem grafischen Indikator, der eigentlichen Benachrichtigung, dem Zeitstempel und einer optionalen Meta-Information⁷¹. Nach der Evaluation des Notification-Centers bzw. der allgemeinen Darstellung von Ereignis-Benachrichtigungen wurde jedoch deutlich, dass auch die jeweilige Instanz, von der aus das Ereignis eingetreten ist, in die Syntax aufgenommen werden sollte. Durch die überarbeitete Syntax, hat der Nutzer so die Möglichkeit, anhand des Notification-Centers eindeutig identifizieren zu können, durch welche Instanz ein Ereignis ausgelöst wurde

7.2.2 Iterative Anpassung der Trigger-Events

Die analytische Evaluation des Trigger-Events führte zu mehreren Anpassungen der erstellten Gestaltungslösungen, um Mängel im Hinblick auf die Gebrauchstauglichkeit zu beheben. Besonders kritisch war dabei die Auswahl der zur Verfügung stehenden Artefakte für einen Trigger, sofern der Nutzer die Trigger-Events-Einstellungen über das „Preferences“-Menü der RCE-Benutzeroberfläche aufruft. In diesem Fall, wurde in den Trigger-Event-Einstellungen keine Vorauswahl getroffen und der Nutzer musste das gewünschte Artefakt anhand des Dropdown-Menüs manuell auswählen. Je nach Kontext eines Nutzers, kann die Anzahl der Elemente in diesem Dropdown-Menü jedoch derart hoch ausfallen, dass dieser sein gewünschtes Artefakt nicht ohne einen erheblichen Aufwand finden kann. Daher wurden die Einstellungen der Trigger-Events angepasst und um eine Vorauswahl der Kategorien erweitert. Der Nutzer kann somit effizienter eine Auswahl des gewünschten Artefaktes vornehmen, wenn dieser zunächst die entsprechende Kategorie wählt.

Im Rahmen eines Cognitive Walkthroughs ließ sich darüber hinaus die Anforderung ermitteln, dass sich mehrere Trigger logisch miteinander verknüpfen lassen sollten, um so komplexere Monitoring-Funktionalitäten zu ermöglichen. Diese Anforderung wurde insbesondere durch den Task „Eine Nachricht schicken lassen, wenn ein Tool nicht verfügbar ist oder einen Wert übersteigt“ identifiziert. Für diese Art des Monitorings ist eine Verknüpfung zweier Trigger erforderlich, die durch ein logisches „oder“ verknüpft

⁷¹ Diese Darstellung kann verschiedenen Mockups des Anhangs entnommen werden

werden müssen. Um dies zu ermöglichen, wurde das Konzept der Trigger-Event-Einstellungen um eine „add Trigger“-Funktionalität innerhalb eines Trigger-Events erweitert.

Eine weitere Anpassung des Trigger-Event-Konzeptes wurde nach einer Absprache einiger Gestaltungslösungen mit Mitgliedern des RCE-Teams durchgeführt. So sahen erste Entwürfe der Trigger-Event-Einstellungen keine Definition der jeweiligen Instanzen vor, die das erstellte Trigger-Event ausführen. Möchte ein Nutzer sich Informationen über ein Trigger-Event z.B. per E-Mail zusenden lassen, während sein eigener RCE-Client nicht aktiv ist, so muss dieser die Möglichkeit haben, die Instanz festzulegen, von der aus das Trigger-Event ausgeführt werden soll. Dieses Szenario wurde zunächst nicht berücksichtigt, weshalb in ersten Gestaltungslösungen keine Auswahl der möglichen Trigger-Instanzen enthalten war.

Während eines Cognitive-Walkthroughs ließ sich zudem eine ähnliche Problematik bei der Erstellung eines Trigger-Events identifizieren. Neben einer Definition der Instanz, die das Ereignis auslösen soll, konnte die Anforderung erhoben werden, dass auch dem entsprechenden Artefakt, auf das ein Trigger angewandt wird, eine Instanz zugeteilt werden muss. So kann unter anderem das Szenario eintreten, dass der Nutzer ein Tool in verschiedenen Workflows einsetzt und für dieses ein Trigger-Event anlegen möchte. Will der Nutzer nun alle Instanzen dieses Tools berücksichtigen oder nur die Instanz eines bestimmten Workflows, so muss er dies in den Trigger-Event-Einstellungen definieren können. Aus diesem Grund erfolgte nochmals eine iterative Überarbeitung des Konzeptes, in dem der Nutzer zu einem Artefakt stets die gewünschte Instanz auswählen kann.

Die soeben erwähnten Iterationen der Anpassung von Gestaltungslösungen, beschreiben im Rahmen dieses Dokumentes lediglich die als besonders kritisch erachteten Aspekte der Gebrauchstauglichkeit, die während der Evaluation des Konzeptes identifiziert wurden. Über diese Iterationen hinaus, fanden viele weitere Anpassungen bestehender Gestaltungslösungen statt, die an dieser Stelle aufgrund des Umfangs nicht näher erläutert werden. Darüber hinaus beschränkten sich diese Anpassungen überwiegend auf weniger kritische Aspekte, wie beispielsweise die Anordnung von Elementen auf der Benutzeroberfläche, missverständliche Symboliken oder ungeeignete Interaktionsmöglichkeiten. Der Entwurf sowie die iterative Überarbeitung von Gestaltungslösungen kann anhand der verschiedenen Mockups sowie Prototypen dem Anhang (Kapitel 9) dieser Ausarbeitung entnommen werden.

Rückblickend betrachtet, konnten anhand der durchgeführten analytischen Evaluation, sowie der Diskussion einzelner Gestaltungslösungen mit Mitgliedern des RCE-Teams, einige kritische Aspekte hinsichtlich der Gebrauchstauglichkeit des erstellten Konzeptes

identifiziert und anschließend iterativ überarbeitet werden. Der Prozess beschränkte sich dabei jedoch nicht allein auf eine abschließende Evaluation gegen Ende, sondern erfolgte über den gesamten Zeitraum, von der Analysephase bis hin zum Abschluss dieses Projektes. Zwar ließ sich im Rahmen der abschließenden Evaluation, durch eine entsprechende Vorbereitung dieser, teils umfangreiches Potenzial zur Steigerung der Gebrauchstauglichkeit identifizieren, allerdings sollten auch die weitaus kleineren Evaluations-Zyklen nicht außer Acht gelassen werden. So ist es im Kontext eines Projektes wie diesem ratsam, bereits in einer frühen Phase der Konzeption Gestaltungslösungen angemessen zu evaluieren, um grundlegende Design-Schwächen nicht erst gegen Ende des Projektes zu identifizieren.

8 Fazit

Die Durchführung des Projektes, von der Analysephase über die Konzeption bis hin zur abschließenden Evaluation stellte eine besondere Herausforderungen dar, die durch eine Betrachtung verschiedener Forschungsgebiete der Mensch-Computer-Interaktion ausführlich behandelt wurde. So konnte die vorgestellte Forschungsfrage, ob es möglich ist, eine einheitliche Identifikation von Anforderungen an Ereignis-Benachrichtigungen für verteilte Systeme vorzunehmen, durch die Erstellung eines als angemessen erachteten Modells adressiert werden. Maßgeblich für dieses Modell, war dabei eine hinreichende Betrachtung verschiedener Konzepte in der Analysephase.

Während der Konzeptphase, ließen sich anschließend einige der in dem Modell identifizierten Anforderungen, in entsprechende Gestaltungslösungen überführen. Aufgrund des knappen zeitlichen Rahmens war es jedoch nicht möglich, alle Anforderungen in der Konzeption eines Benachrichtigungs-Systems zu berücksichtigen, weshalb bislang offen bleibt, in wie fern sich die nicht behandelten Anforderungen auf die verteilte Simulationsumgebung RCE übertragen lassen. Neben einer Identifikation der relevanten Anforderungen an ein gebrauchstaugliches Benachrichtigungs-System, erforderte zudem die Evaluation des entsprechenden Konzeptes eine ausführliche Abwägung verschiedener Methoden aus der Evaluations-Forschung. Grundlegend für eine Vorbereitung eines angemessenen Vorgehens-Modells, ist dabei eine ausführliche Betrachtung, der zur Verfügung stehenden Ressourcen während einer Evaluation. So wurde im Kontext dieses Projektes deutlich, dass sich aus der Vielzahl an verfügbaren Evaluations-Methoden, aufgrund der knappen Ressourcen, nur wenige Ansätze tatsächlich anwenden ließen.

Problematisch dabei war unter anderem, dass die durchgeführte analytische Evaluation zwar durchaus zu wertvollen Erkenntnissen geführt hat, jedoch nicht alle als relevant erachteten Faktoren hinreichend durch diese erfasst werden konnten. Die Konzeption des empirischen Vorgehens-Modells sollte daher für eine ganzheitliche Evaluation nicht außer Acht gelassen werden. Darüber hinaus wurde deutlich, dass sich die einzelnen Prozess-Stufen, von der Analysephase, über die Konzeption bis hin zur Evaluation nur selten klar voneinander abgrenzen lassen. So führten unter anderem neue Erkenntnisse aus der Konzeption des Benachrichtigungs-Systems, zu einer Identifikation neuer Anforderungen, die eine Anpassung des in der Analysephase erstellten Modells bewirkten. Des Weiteren kann auch die Evaluation rückblickend betrachtet, nicht eindeutig als abgeschlossener Prozess bezeichnet werden. Neben der abschließenden Evaluation, fanden so bereits während der Analysephase und zu Beginn der Konzeption viele kleinere Evaluations-Zyklen statt, die ebenfalls zu wertvollen Erkenntnissen

geführt haben und daher in jedem Fall als sinnvoll erachtet werden. Zusammenfassend konnte mit der Durchführung des Projektes und dieser Ausarbeitung eine als hinreichend erachtete Beantwortung der in der Einleitung definierten Forschungsfragen erfolgen.

Die zugrunde liegenden Konzepte, wie das vorgeschlagene Modell der Anforderungen an Ereignis-Benachrichtigungen oder das Vorgehens-Modell der empirischen Evaluation, sind jedoch durch weiterführende Arbeiten im Kontext verteilter Systeme ausführlicher zu verifizieren.

8.1 Ausblick

Als Ausblick an weiterführende Arbeiten sei zunächst das vorgeschlagene Modell zur Identifikation von Anforderungen an Ereignis-Benachrichtigungen in einem verteilten System zu erwähnen. Zwar erfolgte die Erstellung dieses Modell mit dem Fokus, dass sich dieses auch allgemein für verteilte Systeme verschiedener Domänen anbietet, jedoch konnte dies im Rahmen dieser Arbeit nicht verifiziert werden. Fraglich ist dabei insbesondere, ob das vorgestellte Modell eine ganzheitliche Identifikation relevanter Anforderungen beschreibt, oder ob sich noch weitere Anforderungen erheben lassen, die allgemein für verteilte Systeme gültig sind. Da zudem kein vergleichbarer Ansatz zur Festlegung von Anforderungen an Ereignis-Benachrichtigungen in dieser Arbeit berücksichtigt wurden, bleibt zudem bislang unbeantwortet, in wie fern ein solches Modell im direkten Vergleich zu anderen Vorgehensweisen zu einer effizienteren Entwicklung von gebrauchstauglichen Gestaltungslösungen führen kann.

Auch konnten nicht alle der in dem Modell definierten Anforderungen gleichermaßen in dem erstellten Konzept berücksichtigt werden, weshalb eine ganzheitliche Betrachtung dieser, als Ausblick für eine weitere Auseinandersetzung sinnvoll ist. So wirft insbesondere die Einhaltung der Privatsphäre eines Nutzers in einem verteilten System interessante Fragestellungen auf. In wie fern die Privatsphäre eines Nutzers in einem kollaborativen Umfeld zu wahren ist und wie sich dies durch ein verteiltes System realisieren lässt, bleibt im Rahmen dieser Arbeit bislang unbeantwortet. In Bezug auf die Evaluation von gebrauchstauglichen Benachrichtigungs-Systemen, ergeben sich ebenfalls interessante Fragestellungen, die als Ausblick an eine weiterführende Arbeit möglich sind. So bleibt bislang unbeantwortet, ob das in dieser Ausarbeitung vorgeschlagene Vorgehen einer empirischen Evaluation zu einer effizienten Erhebung relevanter Awareness-Aspekte führt und ob sich auch die weiteren als relevant erachteten Faktoren durch dieses Vorgehen adressieren lassen. Hierbei wäre unter anderem eine ausführliche Abwägung der Faktoren gebrauchstauglicher Ereignis-Benachrichtigungen interessant.

Anhang A: Task-Descriptions

Task: Interpretieren, warum ein Workflow unerwartet beendet wurde.

Scope: Notification-Center, Comprehension

Szenario: Ein Workflow in der Durchführung bricht plötzlich unerwartet ab. Der Nutzer möchte nun herausfinden, weshalb dies geschehen ist.

Action-Sequences:

1. Meldung in der Notification-Bar anklicken
 - 1.1. Pfeil-Icon der Notification-Bar anklicken
2. Notification-Center öffnet sich
3. Kategorie der Workflow-Notifications betrachten
 - 3.1. Suchbegriff (Workflowname) in Filtermaske eingeben
 - 3.2. Suche durchführen
4. Identifizieren, ob weitere Informationen der Benachrichtigung verfügbar sind
5. Weitere Informationen der Benachrichtigung durch Klick auf Pfeil-Icon anzeigen
6. Zugehörige Benachrichtigungen öffnen sich
7. Ursprung des Abbruchs ermitteln

Task: Ergebnisse eines Workflows „remote“ anzeigen lassen

Scope: Trigger-Events

Szenario: Ein Nutzer möchte sich anzeigen lassen, wenn einer seiner Workflows erfolgreich beendet wurde. Da er jedoch mit einer anderen Aufgabe „außer Haus“ beschäftigt ist, will er sich per SMS über die Durchführung des Workflows informieren lassen

Action-Sequences:

1. Nutzer öffnet das Kontextmenü des gewünschten Workflows
2. Nutzer wählt über das Kontextmenü den Eintrag „add TriggerEvent“
3. TriggerEvent-Einstellungen mit Vorauswahl des Workflows wird angezeigt
 - 3.1. Nutzer wählt die Trigger-Instanz aus, von der aus der Trigger ausgelöst wird

4. Nutzer wählt unter Trigger die Bedingung „Workflow-Status“
5. Nutzer wählt als möglichen Wert den Status „finished“
6. Nutzer legt als Event „Send SMS“ an
7. Nutzer trägt im nebenstehenden Feld seine Mobilfunk-Nummer ein
8. Nutzer bestätigt seine Eingaben

Task: Informationen über einen Workflow in den eigenen Fokus rücken um Informationen zu erhalten

Scope: Notification-Center, Tagging-System

Szenario: Der Nutzer hat eine neue Aufgabe zugeteilt bekommen, während er ein Monitoring zu einem laufenden Workflow vornimmt. Er möchte daher den laufenden Workflow schnell in seinen Fokus rücken, um sich Informationen zu diesem anzeigen zu lassen

Action-Sequences:

1. Kontext-Menü des gewünschten Workflows auswählen
2. Den Menüeintrag „Tags“ wählen
3. Den vordefinierten „Fokus-Tag“ auswählen

Task: Informationen darstellen lassen, wer das eigens entwickelte Tool einsetzt

Scope: Notification.Center, Trigger-Events

Szenario: Der Nutzer hat ein eigenes Tool in der Simulationsumgebung RCE veröffentlicht. Er möchte nun darüber informiert werden, welche Instanz sein Tool aktuell einsetzt

Action-Sequences:

1. Der Nutzer öffnet das Kontext-Menü des gewünschten Tools
2. Der Nutzer wählt den Menüeintrag „add TriggerEvent“
3. Die TriggerEvent-Einstellungen werden mit der entsprechenden Auswahl geöffnet
4. Der Nutzer legt als Bedingung den Trigger „Tool-Status“ fest
5. Der Nutzer wählt als Wert den Status „running“ aus
6. Der Nutzer fügt über die „add Trigger“-Funktion einen neuen Trigger hinzu (UND)

7. Der Nutzer wählt als weiteren Trigger die Bedingung „External Usage“ aus
8. Der Nutzer wählt als Event aus, dass er eine Benachrichtigung im Notification-Center erhalten möchte
9. Der Nutzer bestätigt seine Auswahl

Task: Für eine eigens definierte Gruppe an Tools, Informationen anzeigen lassen

Scope: Notification-Center, Tagging-System

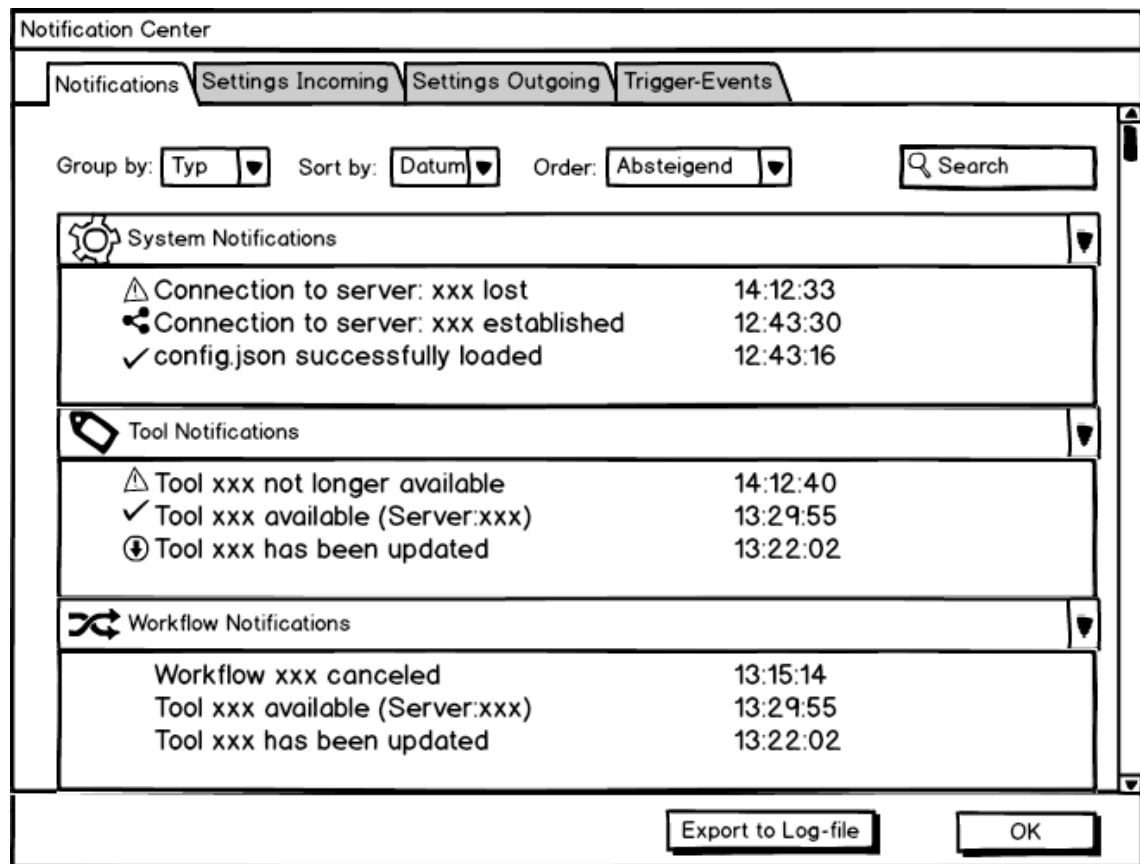
Szenario: Der Nutzer hat in RCE drei eigene Tools integriert und in der verteilten Umgebung veröffentlicht. Er möchte nun, dass er Informationen zu diesen Tools in einer kompakten Darstellung erhält

Action-Sequences:

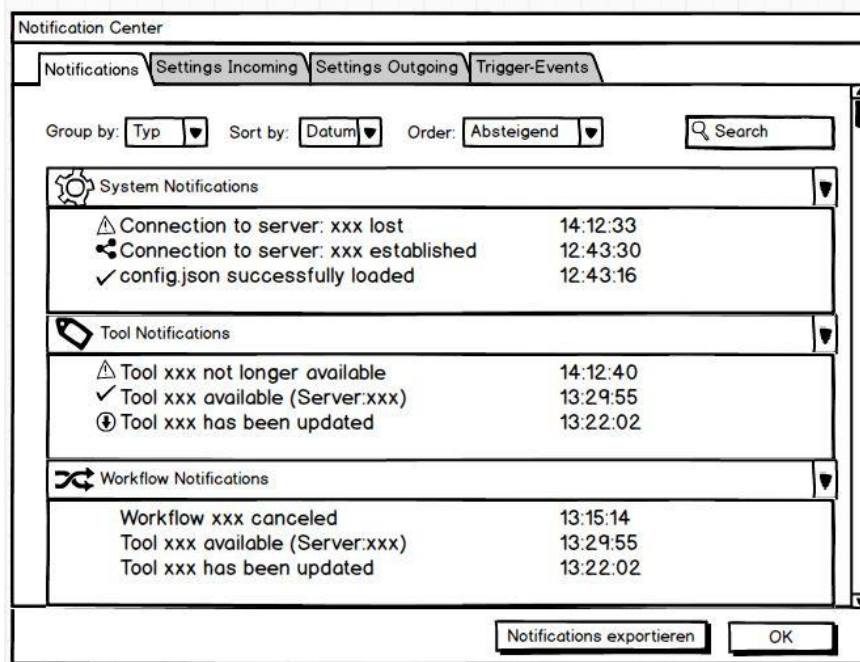
1. Der Nutzer selektiert die drei gewünschten Tools (Multi-Select)
2. Der Nutzer öffnet das Kontext-Menü der gewünschten Tools
3. Der Nutzer wählt den Menüeintrag „Tags“
4. Der Nutzer wählt in dem Submenü den Eintrag „Create new Tag“
5. Das Dialog-Fenster zur Erstellung eines Tags öffnet sich
6. Der Nutzer gibt einen Namen ein und wählt eine gewünschte Farbe
7. Der Nutzer aktiviert die Option „Show in Notification-Center“
8. Das Benachrichtigungs-System erstellt eine entsprechende Notification-Group
9. Der Nutzer öffnet das Notification-Center um Informationen zu seinen Tools zu entnehmen

Anhang B: Balsamiq-Mockups

A.1 Notification-Center



Notification Center				
Sort by: <input type="text" value="Date"/> Order: <input type="text" value="Descending"/> <input type="text" value="Search"/>				
All Notifications	Category	Event	Time	Additional
System Notifications	⚠ System	Connection to server: xxx lost	02.02.2015 - 14:12:33	Shut down till 17:30
Tool Notifications	📥 Tool	Tool xxx has been updated	02.02.2015 - 13:14:15	
Workflow Notifications	🔌 System	Connection to server xxx established	02.02.2015 - 13:07:01	Patato
Settings	🔄 Workflow	Workflow xxx canceled	02.02.2015 - 13:04:33	
	⏸ Workflow	Workflow xxx paused	02.02.2015 - 12:41:11	
	⚠ Workflow	Workflow xxx canceled with errors	02.02.2015 - 12:39:43	
	🔄 Tool	Tool xxx is not longer available	02.02.2015 - 12:39:35	
	✓ System	config.json successfully loaded	02.02.2015 - 10:55:12	



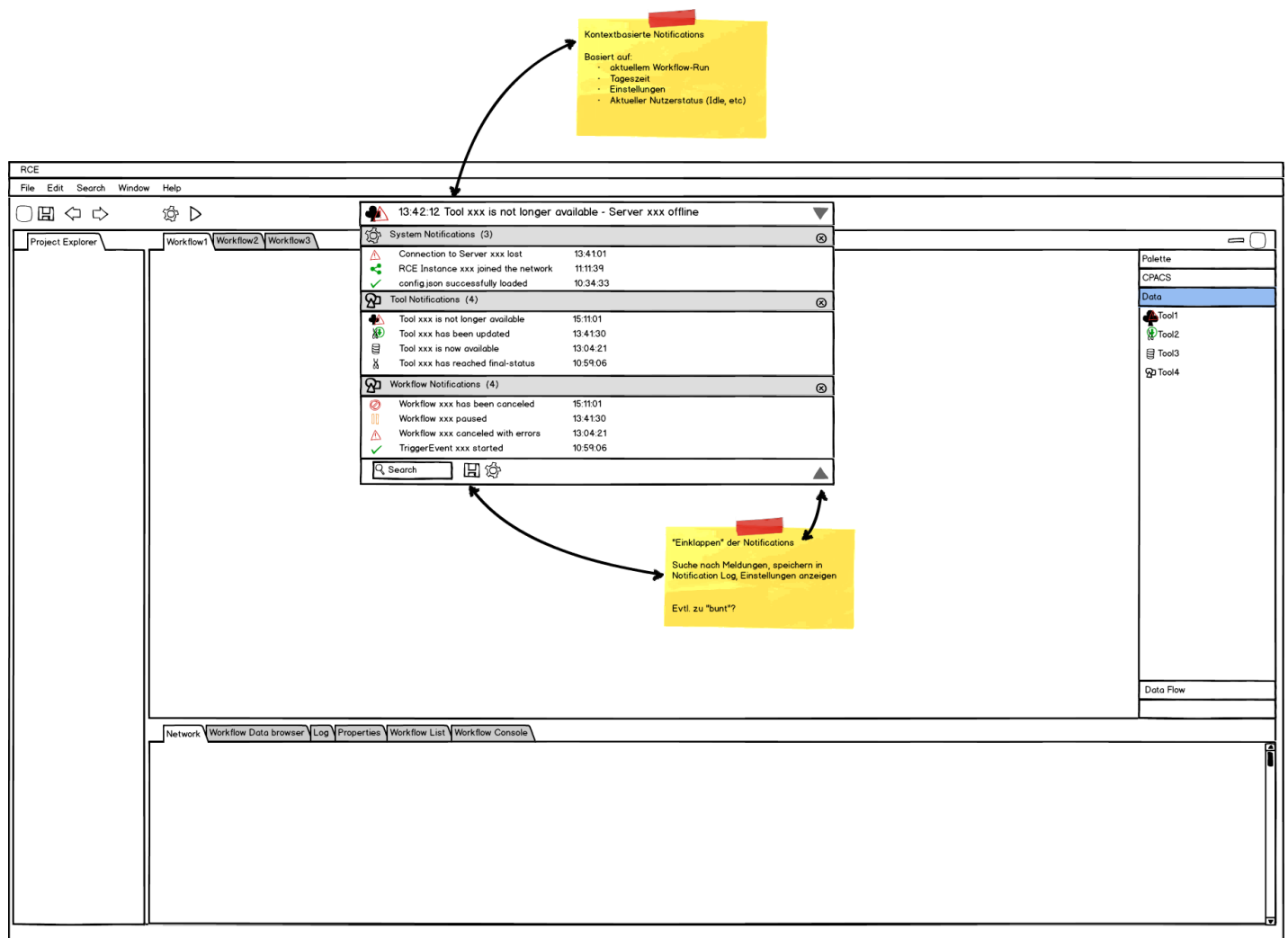
Sortierung der Benachrichtigungen individuell?
Ist eine Klassifizierung der Ereignisse sinnvoll?
-> System, Tool, Workflow, etc.

Filterfunktionen:

- Gruppieren nach
 - Typ: System, Tool, Workflow
 - Priorität: hoch, mittel, niedrig
 - Ohne
- Sortieren nach
 - Datum
 - Bezeichnung

Aufsteigend, absteigend
-Suche

Speicherfunktion in Log-Datei?



Kontextbasierte Notifications

Basiert auf:



- aktuellem Workflow-Run
- Tagesszeit
- Einstellungen
- Aktueller Nutzerstatus (Idle, etc)


"Einklappen" der Notifications


Suche nach Meldungen, speichern in
Notification Log, Einstellungen anzeigen


Evtl. zu "bunt"?


NC_Mockup2 (closed)


 13:42:12 Tool xxx is not longer available - Server xxx offline 


 Tool xxx is not longer available 15:11:01


 Connection to Server xxx lost 13:41:01


 Tool xxx is now available 13:04:21


 System Notifications (3)


 Connection to Server xxx lost 13:41:01


 RCE Instance xxx joined the network 11:11:39

 config.json successfully loaded 10:34:33


 Workflow Notifications (8)


 Workflow xxx has been canceled 15:11:01 Canceled by xxx


 Workflow xxx paused 13:41:30 Paused by xxx

 Workflow xxx canceled with errors 13:04:21 3 related Notifications available

TriggerEvent xxx started 10:59:06

 Show more ...


 Tool Notifications (4)



 Tool xxx is not longer available 15:11:01

Tool xxx has been updated 13:41:30



Tool xxx is now available 13:04:21




Tool xxx has reached final-status 10:59:06

 Search



NC_Mockup2 (collapsed)




 13:42:12 Tool xxx is not longer available - Server xxx offline 



Tool xxx is not longer available
Connection to Server xxx lost
Tool xxx is now available


15:11:01
13:41:01
13:04:21









 System Notifications (3) 



Connection to Server xxx lost
RCE Instance xxx joined the network
config.json successfully loaded

13:41:01
11:11:39
10:34:33

 Workflow Notifications (8)





Workflow xxx has been canceled
Workflow xxx paused
Workflow xxx canceled with errors
TriggerEvent xxx started
TriggerEvent xxx started
Workflow xxx started
Workflow xxx modified
Workflow xxx canceled


15:11:01
13:41:30
13:04:21
10:59:06
10:57:56
07:21:10
07:20:32
06:56:43

Canceled by xxx
Paused by xxx
3 related Notifications available

Canceled by xxx



 Show less. . .

 Tool Notifications (4)



Tool xxx is not longer available

15:11:01



Settings

General

RCE

Help

Eclipse

Notifications

- Notification Focus

- TriggerEvents

- Tags

- Outgoing

Notification Focus

Focus on:

☒ Tags
☐ Current Workflow
☒ Enable Focus

☒ User Messages
☒ Error Messages

Filter:

▶ User Messages

▼ Tags

☒ MeinTag1
☒ MeineWorkflows

▼ Error Messages

☒ Filtertext1, Filtertext1, Filtertext3, ...

Notification Groups

⊕ add Group

⊖ Group Name1

☒ Enable ▼

⊖ System Notifications

☒ Enable ▼

⊖ Tool Notifications

☒ Enable ▼

⊖ Workflow Notifications

☒ Enable ▼

Reset to Default

OK

Notification_handling2 (MetaInfos)

Workflow Notifications (4)			
▶	⊗ Workflow xxx has been canceled	15:11:01	Canceled by xxx
▶	⏸ Workflow xxx paused	13:41:30	Paused by xxx
▼	⚠ Workflow xxx canceled with errors	13:04:21	3 related Notifications available
▶	⚠ Tool xxx is not available	13:04:20	
▶	⚠ Could not reconnect to Server xxx	12:55:54	
▶	⚠ Connection to Server xxx lost	12:40:00	
▶	✓ TriggerEvent xxx started	10:59:06	
▶	⚠ Workflow xxx started	07:21:10	
▶	⚠ Workflow xxx modified	07:20:32	
	Workflow xxx canceled	06:56:43	
▲	Show less. . .		

Settings

General

RCE

Help

Eclipse

Notifications

- General

- Incoming

- Outgoing

- TriggerEvents

Notification Focus

☒ Enable

Focus on:

Bookmarks

Order by:

Time Ascending

Bookmarks:

Workflow1, Workflow3, ..., WorkflowN
Tool3, Tool4, ..., ToolN

Filter tags:

Tag1, Tag2, ..., TagN

Notification Groups

+

 add Group

-

 Group Name1

☒ Enable

Category:

Wichtiges1

Filter:

MeinTag2

+

 add Category

-

 System Notifications

☒ Enable

-

 Tool Notifications

☒ Enable

Reset to Default

OK

NC_Settings Mockup 2 Focus1

Settings

General

RCE

Help

Eclipse

Notifications

- General

- Incoming

- Outgoing

- TriggerEvents

Notification Focus

☒ Enable

Focus on:

Current Workflow

Order by:

Time Ascending

Bookmarks:

Filter tags:

Tag1, Tag2, ..., TagN

Notification Groups

+

 add Group

-

 Group Name1

Category:

All Notifications

+

 add Category

Filter::

Tag1, Tag2, ..., TagN

-

 System Notifications

-

 Tool Notifications

Reset to Default

OK

NC_Settings Mockup2 Focus2

Settings

General

RCE

Help

Eclipse

Notifications

- General

- Incoming

- Outgoing

- TriggerEvents

Notification Focus

☒ Enable

Focus on:

Bookmarks

Order by:

Time Ascending

Bookmarks:

Workflow1, Workflow3, ..., WorkflowN
Tool3, Tool4, ..., ToolN

Filter tags:

Tag1, Tag2, ..., TagN

Notification Groups

+

add Group

-

Group Name1

☒ Enable

Category:

Wichtiges1

Filter:

+

add Category

-

System Notifications

☒ Enable

-

Tool Notifications

☒ Enable

Wichtiges1

MeinTag2

All Notifications

System Notifications

Tool Notifications

Workflow Notifications

Reset to Default

OK

NC_Settings Mockup2 Closed

Settings

General

RCE

Help

Eclipse

Notifications

- Notification Focus

- TriggerEvents

- Tags

- Outgoing

Notification Focus

Focus on: ☒ Focus-Tag ☐ Current Workflow ☒ Enable Focus

☒ User Messages ☒ Error Messages

Filter:

▶ User Messages

▼ Focus-Tag

☒ MeinTag1

☒ MeineWorkflows

▼ Error Messages

☒ Filtertext1, Filtertext1, Filtertext3, ...

Notification Groups

⊕

add Group

⊖

Group Name1

☒ Enable ▼

⊖

System Notifications

☒ Enable ▼

⊖

Tool Notifications

☒ Enable ▼

⊖

Workflow Notifications

☒ Enable ▼

Reset to Default

OK

A.2 Trigger-Events

Notification Center

General
RCE
Help
Eclipse
Notifications
- Incoming
- Outgoing
- TriggerEvents

Search

+ add TriggerEvent

TriggerEvent1

TriggerOn Tool1 ▼

Trigger Tool Status ▼ Tool finished ▼

Event Send E-Mail ▼ name@dlr.de ▼ ▲

TriggerEvent2 - Trigger on: Tool2 Trigger: Time ▼

TriggerEvent3 - Trigger on: Tool1 Trigger: Parameter Range ▼

TriggerEvent4 - Trigger on: Tool2 Trigger: State ▼

Export TriggerEvents OK

Settings

General
RCE
Help
Eclipse
Notifications
- Incoming
- Outgoing
- TriggerEvents

Search

+ add TriggerEvent

TriggerEvent1

TriggerOn Tool1 ▼

Trigger Tool Status ▼ Tool finished ▼

+ add Trigger

Event Send E-Mail ▼ name@dlr.de ▼ ▲

TriggerEvent2 - Trigger on: Tool2 Trigger: Time ▼

TriggerEvent3 - Trigger on: Tool1 Trigger: Parameter Range ▼

TriggerEvent4 - Trigger on: Tool2 Trigger: State ▼

Export TriggerEvents OK

Settings

General

RCE

Help

Eclipse

Notifications

- Incoming

- Outgoing

- TriggerEvents

Search

+

add TriggerEvent

-

TriggerEvent1

TriggerOn

Tool1

▼

Trigger

Tool Status

▼

Tool finished

▼

⊖

Logic:

AND

▼

TriggerOn

Tool1

▼

Trigger

Output X

▼

Please enter Value

+

add Trigger

Event

Send E-Mail

▼

name@dlr.de

▼

-

TriggerEvent2 - Trigger on: Tool2 Trigger: Time

Export TriggerEvents

OK

Settings

General
RCE
Help
Eclipse
Notifications

- Incoming
- Outgoing
- TriggerEvents

Search

+ add TriggerEvent

- TriggerEvent1

Category

Components

TriggerOn

Increase

Trigger Instance

<local>

Trigger

Tool Status

Finished

disabled

Logic:

AND

TriggerOn

Increase

Trigger

Select

Please enter Value

+ add Trigger

Event

E-Mail

Enter Address

- TriggerEvent2 - Trigger on: Tool2 Trigger: Time

Export TriggerEvents

OK

Settings

General
RCE
Help
Eclipse
Notifications
- Incoming
- Outgoing
- TriggerEvents

Search

+ add TriggerEvent

- TriggerEvent1

Category

Components

TriggerOn

Increase

All Instances

Trigger Instance

Server3

Trigger

Value X

4,95

12,5

Logic:

AND

TriggerOn

Increase

Trigger

Select

+ add Trigger

Event

E-Mail

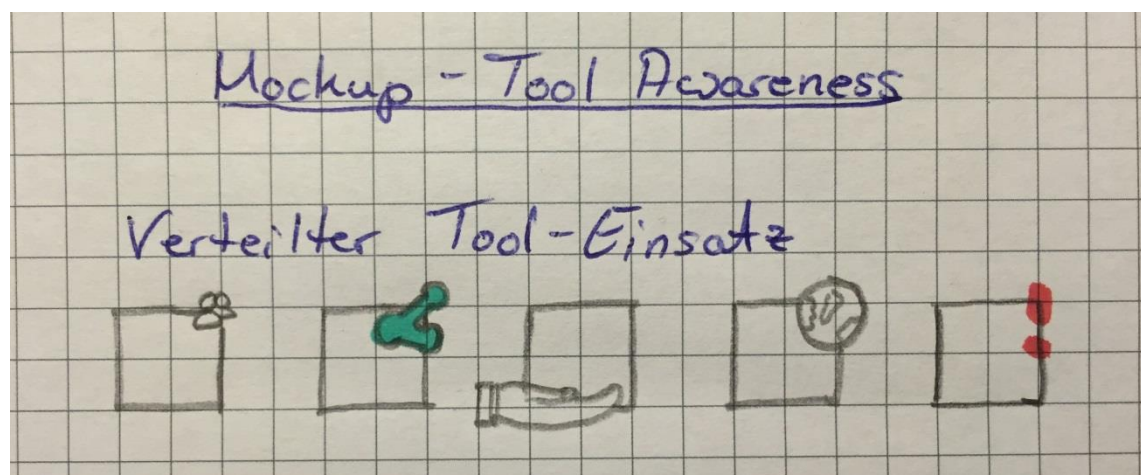
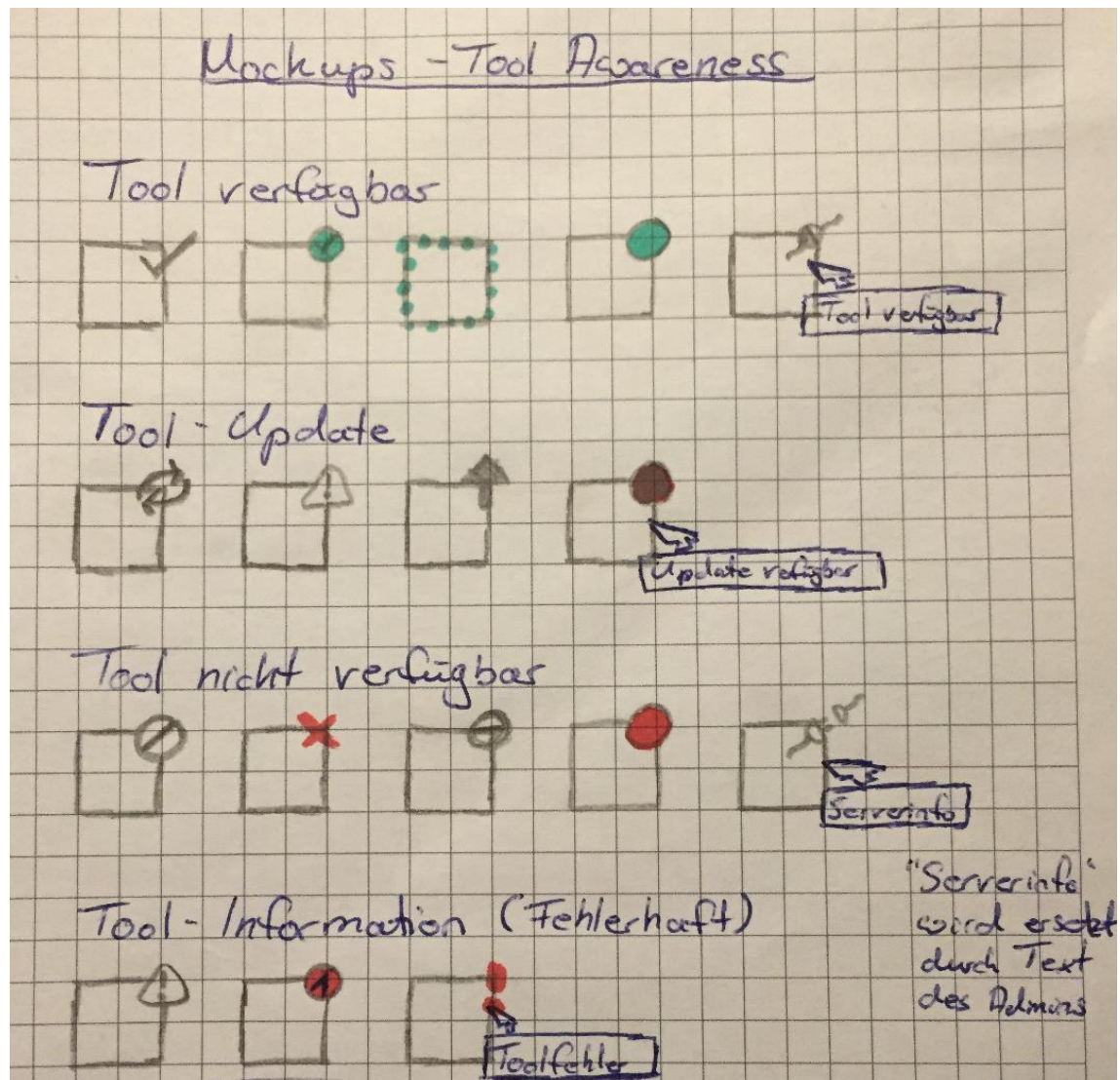
Enter Address

- TriggerEvent2 - Trigger on: Tool2 Trigger: Time

Export TriggerEvents

OK

Anhang B: Paper-based Mockups



Mockup - Benachrichtigung hinzufügen (Use Case)

Art ☒ } Vorauswahl (erforderlich)

Option

Ereignis ☒ } Ereignis (erforderlich)

⊕ Datum

Uhrzeit

⊕ Datum ☒ } Kontextbasiert, je nach Vorauswahl des Users

Uhrzeit

⊕ Neues Ereignis hinzufügen

- Name des Clients
- Dauer des Einsatzes
- Zustand des Tools (Running, Idle, Paused)
- Auslastung des Tools

Filter		Suche	
<input type="text" value="nach Client"/>		<input type="text" value="aufsteigend"/>	
Client	Duration	State	Stress
Seef_ma	10min 20sec	Idle	48% CPU
Seef_ma	2d 4h 43min	Paused	—
Ab_cole	2min 56sec	Running	23% CPU

Name	Logic	Value 1	Value 2
<input checked="" type="checkbox"/> Tool 1			
<input checked="" type="checkbox"/> Tool Status		<input type="text" value="Finished"/>	
<input checked="" type="checkbox"/> Value			
<input type="checkbox"/> X	<input type="text" value="Select"/>	Enter Value	Enter Value
<input checked="" type="checkbox"/> Y	<input type="text" value="AND"/>	3, 0	/
<input type="checkbox"/> Time			
<input type="checkbox"/> Duration			
<input type="checkbox"/> Tool 2			
<input type="checkbox"/> Tool 3			
<input checked="" type="checkbox"/> Tool 4			

Quellenverzeichnis

- Bortz, Döring** (2006): Forschungsmethoden und Evaluation, 4. Auflage.
- McGrenere, Joanna** (2009): Designing effective Notifications for collaborative Development Environments, Department of Computer Science, UBC
- Lambrich, S.** (1999): Microsoft Word 2000 auf einen Blick. Microsoft Press.
- Pinelle, D** (2002): Groupware Walkthrough: Adding Context to Groupware Usability Evaluation, http://hci.usask.ca/publications/2002/groupware_walkthrough.pdf (Datum des Zugriffs: April 2015)
- Institute of Cognitive Science, Wharton C., Rieman J.** (1994): The Cognitive Walkthrough Method: A Practitioner's Guide (Technical Report)
- Bly, Harrison, Irwin** (1993): Media Spaces: Bringing People together in a Video, Audio, and Computing Environment
- Shen H., Sun C.** (2002): Flexible Notifications for Collaborative Systems, Griffith University, Brisbane Australia
- Preece, Rogers, Sharp** (2002): Interaction Design – Beyond Human Computer Interaction.
- Chong J., Siino R.** (2006): Interruptions in Software Teams: A Comparison of Paired and Solo Programmers, Stanford University, Stanford
- Iqbal S., Horvitz E.** (2010): Notifications and Awareness: A Field Study of Alert Usage and Preferences, Microsoft Research, Redmond
- Riihiahho, S.** (2002): The Pluralistic Usability Walk-Through Method, Helsinki University of Technology
- Dantzich M., Robbins D.** (2002): Scope: Providing Awareness of Multiple Notifications at a Glance, Microsoft Research, Redmond
- Schmidt, K.** (2002): The Problem with 'Awareness' – Introductory Remarks on 'Awareness in CSCW', IT University of Copenhagen
- Chignell M., Cordy J.** (2010): The Smart Internet – Current Research and Future Applications, IBM CAS Research, Springer-Verlag
- Booker J., Chewar C.M.** (2004): usability Testing of Notification Interfaces: Are We Focused on the Best Metrics?, Center for Human-Computer Interaction and Department of Computer Science, Blacksburg

Grudin, J. (2001): Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use, Microsoft Research, Redmond

McCrickard D., Chewar M. (2003): Attuning notification design to user goals and attention costs, Communications of the ACM, Vol 46

Mills K. (2003): Computer-Supported Cooperative Work, National Institute of Standards and Technology, Maryland

McCrickard D., Chewar M. (2003): Attuning notification design to user goals and attention costs, Communications of the ACM, Vol 46

Eidesstattliche Versicherung

Name: Szaffenauer

Vorname: Magnus

Matrikel-Nr.: 11068173

Studiengang: Medieninformatik (Master)

Hiermit versichere ich, Magnus Szaffenauer an Eides statt, dass ich die vorliegende Masterarbeit mit dem Titel „Entwurf gebrauchstauglicher Darstellungsformen für eine effiziente Vermittlung von Ereignissen in einer verteilten Simulationsumgebung unter Berücksichtigung nutzerzentrierter Ansätze“ selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der eidesstattlichen Versicherung und prüfungsrechtlichen Folgen (§ 26 Abs. 2 Bachelor-SPO bzw. § 19 Abs. 2 Master-SPO der Hochschule der Medien Stuttgart) sowie die strafrechtlichen Folgen (siehe unten) einer unrichtigen oder unvollständigen eidesstattlichen Versicherung zur Kenntnis genommen.

Auszug aus dem Strafgesetzbuch (StGB)

§ 156 StGB Falsche Versicherung an Eides Statt

Wer von einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

Ort, Datum

Unterschrift